This WACV 2023 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LayerDoc: Layer-wise Extraction of Spatial Hierarchical Structure in Visually-Rich Documents

Puneet Mathur[1,*], Rajiv Jain[2], Ashutosh Mehra[2], Jiuxiang Gu[2], Franck Dernoncourt[2], Anandhavelu N[2], Quan Tran[2], Verena Kaynig-Fittkau[2], Ani Nenkova[2], Dinesh Manocha[1], Vlad I. Morariu[2]
[1]University of Maryland, College Park
[2]Adobe Research
{puneetm, dmanocha}@umd.edu
{rajijain, amehra, jigu, dernonco, anandvn, qtran, kaynigfi, nenkova, morariu}@adobe.com

Abstract

Digital documents often contain images and scanned text. Parsing such visually-rich documents is a core task for workflow automation, but it remains challenging since most documents do not encode explicit layout information, e.g., how characters and words are grouped into boxes and ordered into larger semantic entities. Current state-of-the-art layout extraction methods are challenged by such documents as they rely on word sequences to have correct reading order and do not exploit their hierarchical structure. We propose `LayerDoc`*, an approach that uses visual features, textual semantics, and spatial coordinates along with constraint inference to extract the hierarchical layout structure of documents in a bottom-up layer-wise fashion.* `LayerDoc` *recursively groups smaller regions into larger semantic elements in 2D to infer complex nested hierarchies. Experiments show that our approach outperforms competitive baselines by 10-15% on three diverse datasets of forms and mobile app screen layouts for the tasks of spatial region classification, higher-order group identification, layout hierarchy extraction, reading order detection, and word grouping.*

1. Introduction

Structured documents such as forms, invoices, receipts, resumes, contracts and web/app screen interfaces are ubiquitously used in industry [16] and contain a rich variety of components such as tables, check boxes, widgets, buttons, input fields. Structured documents make use of spatial layout to convey information through potentially nested spatial grouping. However, digital documents (eg. PDF) generally discard most structure and encode only low-level binary information, while document images produced by a scanner or mobile phone scan app are stored in rasterized format (as pixels). Neither of these document formats encode spatial structure explicitly to identify which pieces of text belong together. This leads to challenges for state-of-the-art information extraction techniques, which generally assume that the reading order of text is known [46].

A number of techniques–e.g., LayoutLM [51], LayoutLMv2 [52], DocStruct[47], Form2Seq [1]–model the textual semantics, visual appearance, and spatial location of text to solve sequence labeling and classification tasks. These techniques are able to model spatial information implicitly to assign semantic labels to words, classify a sequence of words (or sub-word tokens), or predict relationships between given regions. However, these methods do not infer the 2D grouping of individual words into semantic elements (e.g., DocStruct assumes candidate regions are provided as preprocessed inputs), nor do they produce the nested structure of a document as output. While LayoutLM is capable of grouping multiple word or sub-word tokens into semantic elements via BIO encoding, the encoding assumes that the reading order of input tokens is correct–but reading order itself is dependent on the structure of the document and is not known, and most OCR systems cannot infer it correctly for complex spatial structure [7].

To illustrate the importance of modeling the structure of a document, consider the example shown in Figure 1. For the use case of digital form authoring, where the goal is to convert a scanned form into a digital format, an algorithm would need to extract characters/words, group them into semantic elements (e.g., a choice label), and further group them into larger elements (a label and the checkbox to its left form a *choice field* element, multiple choice fields form a *choice group*, etc). All of these nested group relationships are important since the labels need to be displayed next to the corresponding checkboxes, and the choice group must consist of mutually exclusive choices that affect the UI, as

*Work done during internship at Adobe Research

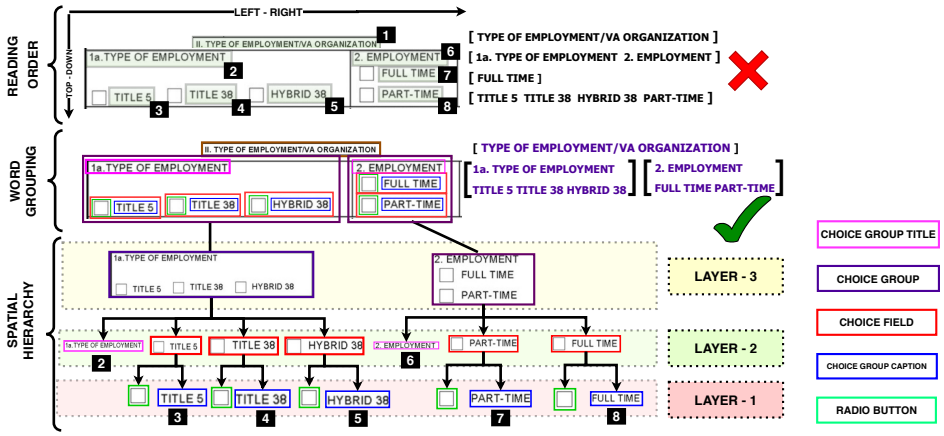


Figure 1: Example of a scanned form document showing the true reading order using numbered black boxes; word grouping based on spatial arrangement; spatial document hierarchy of elements. Reading order extracted naively in a linear fashion (top → down, left → right) is incorrect (top-right). However, the document can be decomposed into a hierarchy, where text fragments group into choice group caption, radio buttons and choice labels grouped into choice fields, etc. `LayerDoc` extracts this spatial hierarchy to group elements and assign them correct semantic labels. The correct reading order is obtained by leaf node traversal of the hierarchy.

checking one box should cause the other boxes in the group to be unchecked. Besides form authoring, other uses of this type of structure include re-flowability across devices [14, 22], adaptive editing of user-interfaces [37], and improving accessibility for user-interactions [54]

Even for other extraction tasks, where the structure itself is not of interest, an understanding of the hierarchical arrangement of text regions is useful for the purpose of producing sequences with accurate reading order. This is important for modern Transformer based language models such as BERT [12] and LayoutLM [51] which depend on the correct order of the input text for downstream tasks and are sensitive to incorrect order [19]. Once the hierarchical structure is extracted as in Figure 1, a traversal of the structure can produce reading order that respects group structure and avoids the errors that OCR algorithms would produce.

We propose **LayerDoc**, a model that uses multimodal deep learning on visual features, textual semantics and spatial geometry as well as constraint inference to generate a complete bottom-up ordered hierarchical arrangement of document layout structure. Within this hierarchy, each node is a rectangular region which is assigned a semantic label, with the leaf nodes consisting of OCR tokens or embedded images. This structure is generated in a layer-wise fashion: given an input set of regions, `LayerDoc` hypothesizes candidate 2D groupings of these regions without the need for IOB tagging, evaluates candidate parent-child links between a child region and parent region (the group it belongs to), then commits to a global parent-child assignment through constraint optimization. The multi-modal nature of `LayerDoc` benefits not only those cases where spatial signals are effective (e.g., where layout based models excel) but also where visual and textual signals are needed, as evident from experiments on diverse datasets of semi-structured forms and scanned user-interfaces. Our novel **contributions**:

1. We propose `LayerDoc` for extracting hierarchical document layout in a layer-wise fashion, recursively grouping smaller spatial regions into larger, semantic elements. We are the first to formulate **nested document hierarchy extraction using transformers**.
2. We propose a multimodal contextual encoder that **maximizes use of context by simultaneously modeling all possible parent-child pairs in a layer**. For element type classification and semantic grouping, this leads to a relative improvement of 10-15% across several metrics.
3. We demonstrate how our **extracted nested hierarchical document structure can improve the inferred token reading order and semantic word grouping** by 8-12%.

2. Related Work

Document layout hierarchy extraction involves two main tasks: spatial element detection and spatial region relationship extraction. Early works [23, 43, 15] used heuristics for both tasks independently, which were later replaced by computer vision models (object detectors) [53, 17, 11, 29] to detect lower-level elements and group them based on spatial overlap. [26] utilized Faster-RCNN[42] for document object detection. Recent 2D transformer-based object detectors such as DETR [5] do not explicitly model the visual hierarchy or leverage multimodal (semantic, spatial and visual) information or contextual modeling. Transformer-based models such as LayoutLM[51], LayoutLMv2 [52], LamBERT [13], DocFormer[3], BROS [18], and TILT [39], have been used for sequence labeling and classification of

spatial regions in documents. However, they do not reason about hierarchy or grouping in an end-to-end fashion. Form2Seq [1] and MMPAN [2] extracted limited types of higher-order structures (Choice Groups, Text Fields and Choice Fields) in form documents. Although Form2Seq utilized a seq2seq network to leverage context, it could not be applied in general settings for end-to-end document spatial hierarchy construction. Recently, DocStruct [47] proposed a multimodal model for extracting parent-child relationships between regions. However, it does not utilize the context of neighboring spatial regions for link prediction, nor does it predict the parent element type, as it is designed for naive key-value pair extraction. Our method uses Transformers to analyze multimodal contextual input from lower-level elements to detect and classify higher-level elements, and reconstruct all layers of the layout hierarchy.

3. Methodology

The document hierarchy is constructed by iteratively grouping elements ("child-boxes") in the current layer into larger regions ("parent-boxes") in the next layer. The child-boxes in the first layer consist of elementary tokens extracted directly from a document page image: textual tokens are extracted by an off-the-shelf OCR system and visual regions (e.g., widgets, radio-buttons, and embedded images in the form use case) predicted by a high-precision object detector. For intermediate layers, our approach hypothesizes a high-recall set of geometrically feasible "potential parent-boxes" directly from the child-boxes, such that each box can group one or more child-boxes and form the next layer in the hierarchy. At the core of our approach is a multimodal model (illustrated in Fig 2 and described in Sec 3.1) that predicts links between a potential parent-box and all of its child-boxes in consecutive layers and jointly predicts the semantic label of the parent box. Not all potential parent-boxes are actual elements, so we use constraint inference to keep the parent-boxes that maximize the child-box link probabilities and satisfy hierarchical constraints. This process is repeated one layer at a time, starting from the lowest layer of elementary tokens and recursively grouping the lower-level elements into higher-level constructs to form a hierarchical arrangement of spatial boxes (see Sec 3.3). We next formalize the problem and provide model details.

Problem Statement: Let I_D represent the input document page of which elementary tokens (OCR, embedded widgets, and icons) and their rectangular bounding boxes are extracted by OCR and a high precision object detector, respectively. The ground truth document hierarchy for a scanned document comprises of spatial boxes b_i, each represented by its coordinates (x_1, y_1, x_2, y_2), where (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates, respectively. Each box has a predefined type label t_i. The textual content (w_i) present in a box is acquired by linearly serializing OCR text tokens lying within the box boundaries. The constituent bounding boxes are arranged in a tree-like format where a box in a higher layer may be a parent of one or more boxes in the layer immediately below it. Thus, each box of the document hierarchy tree contains the list of nested child boxes contained within such that: (i) each child-box is grouped into one and only one parent box i.e. the parent-boxes do not mutually overlap, and (ii) each parent-box groups together all geometrically possible child-boxes within its bounds. Unlike previous works [47, 48], this task does not assume the ground truth parent bounding boxes in each layer to be previously known as part of the input at test time.

3.1. LayerDoc Model

We denote the set of n child boxes serialized in a left-to-right and top-to-bottom fashion in the k^{th} layer as $c_i \in \{c_1, c_2 \cdots c_n\}$ and the j^{th} potential parent box candidate under consideration as p_j. We represent each box with three input modalities: (i) Semantic Cues, (ii) Spatial Cues, and (iii) Structural Cues. We also utilize the visual encoding of the entire scanned document image to augment the spatial and semantic signals with visual cues.

Semantic Cues: Using an off-the-shelf pre-trained language model (*SBert*), we encode the textual content of each box (w_i) into a sentence embedding $s = SBert([[CLS], w_i])$ of dimension $1 \times d_S$, where d_S is the hidden states of pre-trained language model. We concatenate the sentence embedding of the potential parent box s_{p_j} with the sequence of sentence embeddings of child boxes $(s_{c_1}, s_{c_2}, \cdots s_{c_n})$ and pass them through a fully connected layer to form the semantic input sequence $S_j^n = \sigma(W_1([s_{p_j} \oplus s_{c_1} s_{c_2} \cdots s_{c_n}]) + \delta_1)$, where W_1, δ_1, $\sigma(\cdot)$, and $\oplus$ denote the weight matrices, bias, Sigmoid activation function, and concatenation, respectively.

Spatial Cues: We extract the bounding box coordinates to derive the relative layout information of each box. Each bounding box b is represented through its upper-left $([x_1, y_1])$ and bottom-right $([x_2, y_2])$ co-ordinates that are normalized, $b = [\frac{x_1}{W}, \frac{y_1}{H}, \frac{x_2}{W}, \frac{y_2}{H}]$, where H and W are the height and width of the scanned document page. The normalized parent bounding box b_{p_j} is concatenated with the sequence of normalized child bounding boxes $(b_{c_1} b_{c_2} \cdots b_{c_n})$ to form the spatial input sequence $B_n^j = [b_{p_j} \oplus b_{c_1} b_{c_2} \cdots b_{c_n}]$.

Structural Cues: Each child box has a box type t. The parent box type is not known at input. Hence, it is represented by a dummy value of $<$ PBOX $>$ in the input sequence. We concatenate the category types of the parent box followed by the linearly serialized child boxes to obtain the structural input sequence $T_j^n = [<$ PBOX $>: t_{c_1} t_{c_2} \cdots t_{c_n}]$.

Visual Cues: Given the document image I_D, we resize it to a fixed size (h,w,3). It is passed through a visual encoder (VE) to obtain the visual feature map $\eta = VE(I_D)$. We utilize the same input visual feature map across all layers and parent box configurations in a given document.

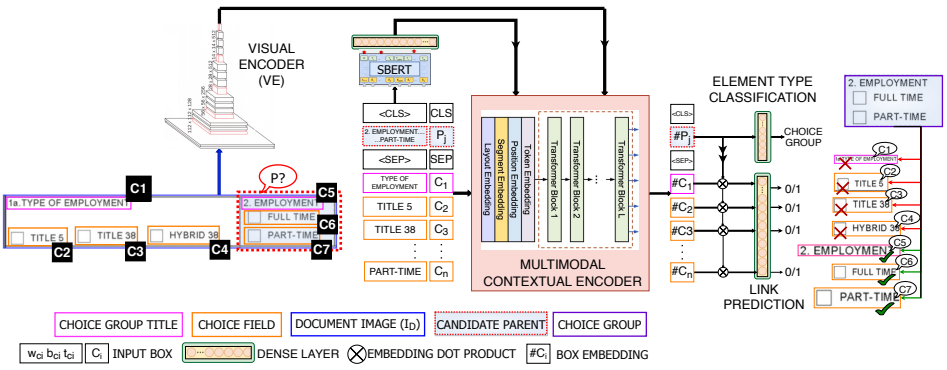


Figure 2: LayerDoc takes raw documents and OCR'ed text as inputs and outputs the spatial hierarchy by grouping lower-level elements into parent boxes, predicting parent-child links and the element type of the parent boxes. The model operates on one layer at a time, considering all child boxes C_i in one layer and candidate parent-boxes P_j in the next layer. The model encodes the visual, textual, and spatial features of each element into a sequence to compute an encoding, which is then used for link prediction and element type classification. Candidate parents are generated using the child-boxes C_i, the final parent set is selected via constraint optimization, and the model is applied recursively to build the hierarchy bottom-up.

Multimodal Contextual Encoder (Λ) combines the structural, spatial and visual cues extracted from the input potential parent box and the sequence of child-boxes through a Transformer-based language model. The semantic cues are concatenated with the embedding of each input box using late fusion as denoted by $\bigoplus$ due to the limitation dictated by the LayoutLM backbone. The final box embedding sequence is $X_j^n = \Lambda([B_j^n; T_j^n; \eta] \bigoplus S_j^n)$. The box embedding sequence X_j^n is matrix multiplied with the parent box embedding $X_j^n[:p_j]$ as $X_j^n[:p_j] \otimes X_j^n$ vector, where $\otimes$ means matrix multiplication. This results in a dot product of each child box embedding with parent box embedding to obtain $[\hat{p}_j; \hat{c}_1, \hat{c}_2, \cdots \hat{c}_n]$.

Link Prediction and Element Type Classification: The child box representations ($[\hat{c}_1, \hat{c}_2, \cdots \hat{c}_n]$) are passed through a dense fully-connected layer followed by a *Sigmoid* layer to generate the link probabilities between each child box c_i and a potential parent box p_j: $\alpha_{j1}, \cdots \alpha_{jn} = \sigma(W_2([\hat{c}_1, \cdots \hat{c}_n]) + \delta_2)$, where W_2, δ_2 and $\sigma(\cdot)$ are the weight matrices, bias and Sigmoid activation function, respectively. The parent box representation ($\hat{p}_j$) is passed through a dense fully-connected layer followed by a *softmax* to predict element type $\varphi_j = \sigma(W_3(\hat{p}_j) + \delta_3)$.

3.2. Training LayerDoc

Negative Parent Sampling: Most potential parent-boxes will be false positives, so to deal with the sparsity of positive samples at test time, we introduce negative sampling [36] in the training regime inspired by [48, 47]. For each training sample having at least one positive link between the potential parent-box and any of the input child-boxes, we add an unrelated parent-box example to the training set for the same setting to make the training robust to negative samples.

Multi-task Training: Element type classification uses a weighted cross-entropy loss to adjust for class imbalance, while link prediction uses a negative sample cross-entropy loss [48] to account for negative data augmentation. Both tasks are correlated and reinforce each other, so we use multi-task training to optimize both tasks simultaneously. The final optimization uses a weighted sum of the link prediction loss and element classification loss $L = \lambda L_{Link} + (1-\lambda) L_{Class}$, where the weighting factor λ is a hyperparameter.

3.3. Inferring Document Layout Hierarchy

We recursively group child-boxes into parent-boxes such that the parent-boxes of the k^{th} layer become child-boxes of the $k+1^{th}$ layer, iterating until only one parent-box remains. Each iteration involves three steps: (i) parent-box candidate generation, (ii) candidate link prediction and type classification, and (iii) constraint inference. The first iteration uses elementary token boxes t_i (OCR text, widgets, icons, etc) as child-boxes. Step (i) hypothesizes geometrically feasible potential parent-boxes (m candidates with an upper limit of $O(n^4)$ due to all relevant combinations of box co-ordinates) ensuring a high-recall collection of potential parent-boxes. Step (ii) predicts parent-child links and element types for each candidate parent-box with all the child-boxes as input, returning link probabilities $\alpha_{ji} \forall i \leq n; j \leq m$. Step (iii) selects the subset of parent boxes that are mutually non-overlapping, cover all child-boxes, and maximize the constraint optimization function described next.

Parent Box Proposal are created by utilizing the geometric constraints of the child boxes. We obtain sets of horizontal (x_{min}, x_{max}) and vertical (y_{min}, y_{max}) coordinates from the child box coordinates and merge them if lying within a threshold distance of each other to cluster closely placed

coordinates and reduce the search space of coordinates. We choose two coordinate points from both x and y sets to form one rectangular parent box.

Constraint Inference: For the k^{th} layer, LayerDoc predicts link probabilities α_{ij} between each pair of potential parent box p_i and child box c_j. The best set of parent boxes is selected by solving a constraint optimization problem, maximizing the cost function $\hat{Y} = \max_{y_i \in \Upsilon} \sum_i^m \omega_i y_i$, where $\omega_i = \kappa + \frac{\sum_j \hat{\alpha}_{ij}}{Ar(p_i)^k / Ar(\cup_{j=0}^n c_j)}$, $\hat{\alpha}_{ij}$ is the adjusted link probability between p_i and c_j such that $\hat{\alpha}_{ij} = \alpha_{ij} - 1$. κ is a large constant added to make all parent scores positive to avoid trivial solution of all weights as zero. $Ar(.)$ defines area of a box and $(\cup_{j=0}^n c_j)$ is the union of all n child boxes. $y_i \to 1$ represents the case where the potential parent box p_i is accepted as a valid parent box. The optimization is subject to constraint space $\Upsilon = \Upsilon_1 \cap \Upsilon_2$ defined over the set of all pairs of potential parent boxes $\Re^m$, where $\Upsilon_1 : y_i \in \{0, 1\}$ and $\Upsilon_2 : y_a + y_b \leq 1 \mid \forall\ a, b \in \Re \times \Re,\ Ar(p_a \cap p_b) > 0$. This is a typical Maximum Independent Set Problem [45] when reduced to a simple linear-programming relaxation by constraining y_i to be binary. It can be solved using Integer Linear Programming (ILP). However, the number of parent boxes grow exponentially, forcing us to further relax the ILP solution by greedily selecting one parent with highest ω_i at a time, leaving improved solutions to future work.

4. Experiments

Datasets: We train and test the `LayerDoc` model on three datasets, Hierarchical Forms, RICO and FUNSD, which provide scanned document images as input. The data statistics and class labels are given in Appendix B Table 6 and 7.

(1)Hierarchical Forms [1] is a rich corpus of scanned form documents from diverse domains like insurance, finance, and government agencies. The documents are human-annotated with labelled bounding boxes, element type, and element relations for a set of 14 constituent elements such as Text Fields, Checkboxes, Choice Groups, Widgets, Tables, Image, Header, Footer, etc. **(2) RICO**[10] is a dataset of more than 66k layout hierarchies of mobile app screens augmented with semantic annotations of UI components. The bounding boxes, element labels and nested hierarchies are from app source code. **(3) FUNSD** [21] is a dataset of noisy scanned forms with shallow hierarchies and filled form fields.

Training: We experiment with four ablation settings using LayoutLM [51] (LayerDoc$_{LLM}$) or LayoutLMv2 [52] (LayerDoc$_{LLMv2}$) in multimodal context encoder. LayoutLMv2 extracts visual cues via Detectron2 [50]. We experiment with and without SentenceBERT [41] for extracting semantic cues (LayerDoc$_{LLM+SBERT}$ and LayerDoc$_{LLMv2+SBERT}$). We use an equally balanced train-validation split. **Object detector**: We utilize Faster-RCNN trained on the training set of Forms/RICO/FUNSD dataset to infer lower-level elementary tokens such as widgets, images, etc. **Box Types** of elementary boxes are obtained object detector predictions.(See Appendix F).

Evaluation Tasks: We evaluate LayerDoc on five tasks: Element Type Classification and Group Identification for specific components of the architecture; Element Detection, Reading Order and Grouping for full hierarchy. **Element Type Classification:** Evaluates the parent-box type classification using weighted F1-score for each type, using ground truth child-boxes as input at test time. **Group Identification:** Evaluates link prediction between the candidate parent-box and child-boxes using macro F1 score using ground-truth child-boxes as input at test time. **Hierarchy Reconstruction:** Elementary tokens (words+bounding boxes) are given as input and all other layers use the predictions from the previous layer. We evaluate document layout hierarchy predicted in Sec 3.3 using Mean Average Precision (mAP) (0.5 IoU threshold) between ground truth and predicted bounding boxes using the standard teacher forcing technique [49]. We also utilize the Adjusted Rand-index [40] to measure the similarity between two hierarchies in each layer as well as for the whole layout hierarchy in aggregate. We consider the child-boxes in a given layer linked to the same parent-box as one cluster and consider the predicted parent-boxes to match if the ground truth if IoU > 0.5. **Reading Order Comparison**: Following [48], we sort the predicted layout hierarchy and traverse the bounding boxes in-order to recover the sequence of OCR tokens. We then compare the predicted reading order sequence against the ground truth reading order using Average Page-level BLEU (p-BLEU) and Average Relative Distance (ARD) [46]. Additional details can be found in the Appendix C. **Grouping Comparison:** We evaluate the word and element grouping. Similar to [24, 46], we utilize the word grouping metric to calculate the F1, precision and recall of intervals in the predicted word sequence belonging to an element compared to the ground truth sequence.

5. Results and Analysis

We present our experimental results, where **bold** in tables denotes the best performing model. Colored text represents the proposed `LayerDoc` with LayoutLMv2 backbone and SentenceBERT for semantic cues. Values not reported by the baseline models are indicated by (–) dashes.

5.1. Element Type Classification

Hierarchical Forms: Table 1a shows element classification where we compare `LayerDoc` with MFCN [53], DLV3+ [38], Form2Seq [1] as they report strong baseline performance for this task. Form2Seq is a competitive baseline that uses seq2seq modeling of spatial regions for element classification and extraction. However, it struggles to handle long-range dependencies in dense forms with large sequences of tokens. MFCN and DLV3+ are strong convolution based

	Modality	Model	TableRow	ChoiceGroup	Footer	Section	ListItem	Table	TextRun	TableCell	TextBlock	List	Field	Header	Overall
Baseline	Visual	MFCN [53]	–	0.0	–	0.71	0.54	–	0.11	–	0.46	0.90	–	0.69	–
	Visual	DLV3+ [38]	–	0.57	–	0.55	0.75	–	0.69	–	0.86	0.48	–	0.83	–
	Spatial + Text	Form2Seq [1]	–	0.78	–	0.67	0.90	–	0.85	–	0.91	0.93	–	**0.85**	–
Ablation	Spatial	LayerDoc$_{LLM}$	0.36	0.74	0.65	0.57	0.74	0.20	0.61	0.57	0.41	0.35	0.42	0.15	0.43
	Spatial + Text	LayerDoc$_{LLM+SBERT}$	0.48	0.68	0.75	0.67	0.79	0.29	0.74	0.89	0.84	0.41	0.80	0.72	0.76
	Spatial + Visual	LayerDoc$_{LLMv2}$	0.69	0.89	0.90	0.76	0.94	**0.96**	0.82	0.97	0.97	0.93	0.94	0.35	0.89
	Spatial + Visual + Text	LayerDoc$_{LLMv2+SBERT}$	**0.92**	**0.90**	**0.92**	**0.86**	**0.96**	0.94	**0.88**	**0.98**	**0.98**	**0.95**	**0.94**	0.67	**0.90**

(a) Hierarchical Forms Dataset

	Modality	Model	List Item	Text	Checkbox	TextButton	Modal	Toolbar	Card	Drawer	Multi-Tab	WebView	Input	Button Bar	Tile	Overall
Baseline	Visual	Faster-RCNN [42]	0.55	0.54	0.29	0.36	0.48	0.63	0.18	0.61	0.45	0.45	0.11	0.03	0.48	0.48
	Visual	UIED [6]	0.62	0.61	0.35	0.41	0.62	0.83	0.27	0.74	0.51	0.45	0.19	0.10	0.60	0.71
	Visual	DETR [5]	0.67	0.65	0.39	0.46	0.67	0.86	0.30	0.75	0.52	0.48	0.20	0.12	0.63	0.72
	Spatial + Visual + Textual	LayoutLMv2 [52]	0.82	0.72	0.39	0.50	0.73	0.88	0.42	0.78	0.55	0.61	0.16	0.18	0.67	0.75
Ablation	Spatial	LayerDoc$_{LLM}$	0.80	0.74	0.42	0.51	0.69	0.94	0.35	0.81	0.60	0.53	0.22	0.18	0.68	0.76
	Spatial + Text	LayerDoc$_{LLM+SBERT}$	0.82	0.74	0.46	0.53	0.59	0.94	0.45	0.83	0.61	0.56	0.20	0.70	0.68	0.78
	Spatial + Visual	LayerDoc$_{LLMv2}$	0.87	**0.77**	0.46	0.53	**0.78**	0.93	0.46	0.86	0.59	0.65	**0.28**	0.2	0.68	0.79
	Spatial + Visual + Text	LayerDoc$_{LLMv2+SBERT}$	**0.88**	0.76	**0.47**	**0.55**	0.65	**0.96**	**0.49**	**0.87**	**0.71**	**0.68**	0.20	**0.78**	**0.73**	**0.80**

(b) RICO Dataset

Table 1: Results comparing F1 scores of LayerDoc with baselines and ablative components for **element classification task** for label-wise and overall spatial elements in (a) **Hierarchical Forms** and (b) **RICO dataset**. Our proposed approach outperforms the baselines, and ablation analysis shows that each individual component contributes to the overall performance.

	Modality	Model	TableRow	ChoiceGroup	Footer	Section	ListItem	Table	TextRun	TableCell	TextBlock	List	Field	Header	Overall
Baseline	Visual	MFCN [53]	–	0.28	–	–	–	–	–	–	–	–	0.19	–	–
	Visual	DLV3+ [38]	–	0.47	–	–	–	–	–	–	–	–	0.51	–	–
	Spatial + Text	Form2Seq [1]	–	0.61	–	–	–	–	–	–	–	–	0.86	–	–
	Spatial + Text + Visual	MMPAN [2]	–	0.63	–	–	–	–	–	–	0.88	–	0.90	–	–
	Spatial + Text + Visual	DocStruct [47]	0.39	0.20	0.18	0.21	0.16	0.09	0.28	0.40	0.27	0.14	0.30	0.07	0.36
Ablation	Spatial	LayerDoc$_{LLM}$	0.36	0.64	0.45	0.28	0.20	0.14	0.44	0.41	0.51	0.22	0.38	0.38	0.41
	Spatial + Text	LayerDoc$_{LLM+SBERT}$	0.38	0.68	0.51	0.48	0.33	0.68	0.52	0.78	0.65	0.45	0.56	0.45	0.55
	Spatial + Visual	LayerDoc$_{LLMv2}$	**0.90**	0.75	0.75	**0.78**	0.82	**0.83**	0.49	0.76	0.90	0.75	0.85	0.15	0.79
	Spatial + Visual + Text	LayerDoc$_{LLMv2+SBERT}$	0.85	**0.78**	**0.80**	0.67	**0.85**	0.70	**0.79**	**0.82**	**0.92**	**0.77**	**0.92**	**0.49**	**0.81**

(a) Hierarchical Forms Dataset

	Modality	Model	List Item	Text	Checkbox	TextButton	Modal	Toolbar	Card	Drawer	Multi-Tab	WebView	Input	Button Bar	Tile	Overall
Baseline	Visual	Faster-RCNN [42]	0.20	0.24	0.29	0.36	0.28	0.31	0.15	0.21	0.25	0.18	0.21	0.15	0.35	0.27
	Visual	UIED [6]	0.24	0.35	0.45	0.40	0.32	0.48	0.27	0.56	0.49	0.55	0.69	0.50	0.56	0.52
	Visual	DETR [5]	0.32	0.39	0.49	0.45	0.38	0.54	0.33	0.61	0.55	0.62	0.72	0.54	0.59	0.55
	Spatial + Visual + Textual	LayoutLMv2 [52]	0.77	0.80	0.69	0.75	0.42	0.83	0.72	0.69	0.81	0.82	0.81	0.74	0.69	0.72
Ablation	Spatial	LayerDoc$_{LLM}$	0.25	0.40	0.52	0.40	0.44	0.52	0.25	0.33	0.40	0.63	0.36	0.53	0.40	0.50
	Spatial + Text	LayerDoc$_{LLM+SBERT}$	0.26	0.41	0.54	0.42	0.44	0.68	0.28	0.35	0.45	0.81	0.39	0.55	0.58	0.55
	Spatial + Visual	LayerDoc$_{LLMv2}$	0.81	0.86	0.75	0.80	0.45	0.88	0.77	0.76	0.86	0.86	0.87	0.81	0.74	0.86
	Spatial + Visual + Text	LayerDoc$_{LLMv2+SBERT}$	**0.83**	**0.88**	**0.77**	**0.82**	**0.47**	**0.90**	**0.79**	**0.77**	**0.88**	**0.88**	**0.96**	**0.84**	**0.76**	**0.89**

(b) RICO Dataset

Table 2: Results comparing F1 scores of LayerDoc with baselines and ablative components for **group identification task** for label-wise and overall spatial elements in (a) **Hierarchical Forms** and (b) **RICO dataset**. Our proposed approach outperforms the baselines, and ablation analysis shows that each individual component contributes to the overall performance.

Model	Element Classification (F1)	Group Identification (F1)
BERT [21]	0.64	0.29
GNN + MLP [4]	0.64	0.39
UniLMv2-large [3]	0.70	–
SPADE [20]	0.71	0.41
StrucTexT [28]	0.83	0.44
LayoutLMv1-large [51]	0.78	0.42
FUDGE [9]	0.66	0.56
SERA [55]	–	0.65
BROS [18]	0.81	0.66
MSAU-PAF [8]	0.83	0.75
LayoutLMv2-large [52]	0.84	–
DocFormer-large [3]	0.84	–
LayerDoc$_{LLMv2+SBERT}$ **(Ours)**	**0.86**	**0.78**

Table 3: Comparison of LayerDoc (w/ LayoutLMv2 and SentenceBert) with baseline models for **element type classification (entity labeling)** and **group identification (linking)** on the **FUNSD** dataset. LayerDoc outperforms all recent top-performing systems in terms of F1 score.

baselines utilized specifically in the document understanding domain. All three baselines were designed to work for a limited set of elements found in the lowest layers of the hierarchy, preventing comparison between all element types.

RICO: Table 1b reports results for RICO dataset. We establish a strong baseline UEID [6] that uses a mix of text detector and traditional computer vision techniques to classify and extract spatial elements. Inspired by [25], we compare Faster-RCNN[42] which is a traditional object detector. We also fine-tune and evaluate recent transformer based object detection models such as DETR [5] and Swin Transformer [30] on UI interfaces from RICO dataset. Visual object detectors are not able to leverage semantic context necessary for document understanding. LayoutLMv2[52] model utilizes visual, spatial as well as semantic context. However, it is pre-trained for language modeling tasks as opposed to layout hierarchy extraction objective. **Performance of LayerDoc** with LayoutLMv2 backbone and SentenceBERT shows significant gains across all element types as it benefits from contextual modeling of spatial regions, multimodal input to the contextual encoder, and multi-tasking objective aimed at optimizing the element type classification and group identification simultaneously. *Header* type elements in Hierarchical Forms dataset are an exception where our model underperforms the Form2Seq baseline. Lower performance of header can be attributed to model overfitting as the header class is a minority in the dataset. LayerDoc is trained to predict several different components simultaneously as opposed

Dataset	Model	Reading Order		Word Grouping		
		p-BLEU (↑)	ARD (↓)	P	R	F1
FUNSD	Heuristics	0.69	8.46	–	–	–
	LayoutLMv1 [51]	0.89	2.54	0.82	0.88	0.85
	LayoutLMv2 [52]	0.92	2.21	0.84	0.87	0.86
	LayoutReader [46]	0.98	1.75	–	–	–
	ROPE [24]	–	–	0.88	0.90	0.89
	LayerDoc$_{LLM}$	0.98	1.68	0.82	0.79	0.80
	LayerDoc$_{LLM+SBERT}$	0.99	1.65	0.85	0.90	0.87
	LayerDoc$_{LLMv2}$	0.98	1.63	0.86	0.92	0.89
	LayerDoc$_{LLMv2+SBERT}$	**0.99**	**1.60**	**0.92**	**0.93**	**0.92**
RICO	Heuristics	0.49	1.77	–	–	–
	Faster-RCNN [42]	0.55	1.76	0.45	0.76	0.57
	UEID [6]	0.61	1.75	0.45	0.76	0.57
	DETR [5]	0.63	1.74	0.48	0.79	0.60
	LayoutLMv2 [52]	0.65	1.72	0.63	0.83	0.72
	LayerDoc$_{LLM}$	0.65	1.70	0.68	0.95	0.79
	LayerDoc$_{LLM+SBERT}$	0.67	1.68	0.68	0.94	0.79
	LayerDoc$_{LLMv2}$	0.69	1.62	0.73	0.95	0.83
	LayerDoc$_{LLMv2+SBERT}$	**0.70**	**1.60**	**0.77**	**0.97**	**0.87**

Table 4: Results for **reading order** (p-BLEU and ARD) and **word grouping** (F1, P, R) for **FUNSD and RICO** dataset.

Dataset	Model	Hierarchy Reconstruction	Rand-index Test		
		mAP (↑)	P	R	Fi
FUNSD	LayoutLMv1 [51]	0.27	0.51	0.54	0.52
	LayoutLMv2 [52]	0.35	0.61	0.62	0.61
	LayerDoc$_{LLM}$	0.45	0.77	0.72	0.74
	LayerDoc$_{LLM+SBERT}$	0.48	0.72	0.81	0.76
	LayerDoc$_{LLMv2+SBERT}$	**0.50**	**0.78**	**0.83**	**0.80**
RICO	Faster-RCNN [42]	0.15	0.32	0.33	0.33
	UEID [6]	0.21	0.35	0.48	0.39
	DETR [5]	0.21	0.43	0.48	0.45
	LayoutLMv2 [52]	0.23	0.55	0.54	0.55
	LayerDoc$_{LLM}$	0.19	0.70	0.74	0.72
	LayerDoc$_{LLM+SBERT}$	0.22	0.74	0.73	0.74
	LayerDoc$_{LLMv2+SBERT}$	**0.27**	**0.86**	**0.84**	**0.85**
Hierarchical Forms	DocStruct [47]	0.10	0.35	0.36	0.36
	LayerDoc$_{LLM}$	0.10	0.33	0.51	0.40
	LayerDoc$_{LLM+SBERT}$	0.11	**0.36**	0.51	**0.42**
	LayerDoc$_{LLMv2+SBERT}$	**0.12**	0.31	**0.55**	0.40

Table 5: Results for **hierarchy reconstruction** (mAP and Rand-index test) for **FUNSD, Hierarchical Forms and RICO** dataset.

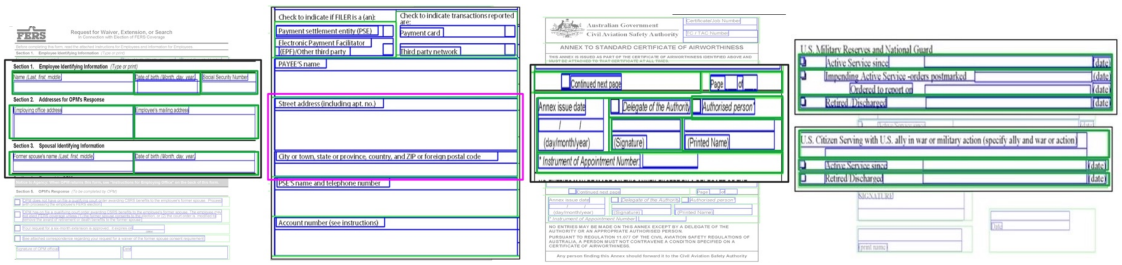
to Form2Seq and DLV3+ baselines which are specifically trained on selective components. Moreover, visual modality does not help element type prediction as headers are usually localized in a small part of the document and do not benefit from contextual modeling.

5.2. Group Identification

Hierarchical Forms: Table 2a shows group identification results where we compare against image segmentation baselines - DeepLabV3+ and MFCN for element extraction. These models often make mistakes in case of closely spaced text blocks and text fields, struggling to predict complete choice fields and choice groups due to their inability to capture complete horizontal context. Form2Seq[1] and MMPAN [17] baselines use LSTM-based seq2seq models to extract multimodal hierarchical associations. We consider the settings where ground truth is given as input to the next step of the pipeline. Results for DeepLabV3+, MFCN, Form2Seq, and MMPAN are derived from [1] which evaluated them to work with specific inputs (text blocks) and to give certain outputs (text blocks, choice groups, choice fields), hence the sparsity in their results. We additionally evaluate DocStruct [47] on Hierarchical Forms, a recent state-of-the-art method for layout structure extraction by re-implementing it for generic semi-structured documents. **RICO:** We evaluate the task of group identification on RICO using hybrid deep networks (UIED), traditional (Faster-RCNN) as well as Transformer-based DETR for 2D object detection baselines. The input to the model is the raw document image while the outputs are predicted bounding boxes with class labels. LayoutLMv2 model is fine-tuned and evaluated similarly to [27]. **Performance of `LayerDoc`**: `LayerDoc` is significantly better compared to all baselines by a large margin for Hierarchical Forms, except for *Choice Fields*. Form2Seq and MMPAN outperform in grouping *text blocks* and *widgets* into *choice field* elements as they were designed to selectively handle such elements. DocStruct severely underperforms against `LayerDoc` on complex hierarchical forms due to the lack of document-level context and inability to generalize beyond simple key-value pair elements. For RICO, both Faster-RCNN and DETR are weaker than `LayerDoc` as they do not leverage multimodal input. `LayerDoc` outperforms LayoutLMv2 due to its superior recursive parent-child link prediction approach. **Performance on FUNSD:** Table 3 compares multiple state-of-the-art methods for element classification and group identification on FUNSD. `LayerDoc` outperforms all other models on extracting and classifying key-value pairs in noisy forms. **Ablation Study:** We denote a darker green shade to indicate better F1 performance, and ablation is indicated by the "Modality" column across the tables. We observe a consistent benefit of using both visual as well as textual modalities in `LayerDoc` across all tasks. Visual cues extracted by Detectron2 in LayoutLMv2 backbone improves performance as most semi-structured documents have visually rich elements such as tables, check boxes, widgets, buttons, input fields. Semantic cues help improve identification of most elements, except *table* and *sections* elements as they rely more heavily on spatial boundaries and neighbouring white spaces for accurate extraction.

Hierarchy Reconstruction: We evaluate the predicted document layout hierarchy in Table 5. Elementary tokens (words+bounding boxes) are input and each layer uses the predictions from the previous layer in a recursive manner. Unlike past hierarchy extraction techniques applied to FUNSD [47], we do not assume ground truth parent boxes to be a part of the input during hierarchy inference. We evaluate using Mean Average Precision (mAP) of predicted boxes with a 0.5 IoU threshold between ground truth and predicted bounding boxes. To generate hierarchies from baseline models, we use the elements detected at inference to arrange them in a bottom-up hierarchy based on geometric constraints. We show that `LayerDoc` with LayoutLMv2 and SentenceBERT outperforms other configurations on all three datasets where ablations show the usefulness of visual, spatial and textual cues.

Reading Order: Table 4 compares reading order of OCR tokens based on the extracted layout hierarchy. We implement a heuristics baseline that linearly sorts the words from left to right and top to bottom based on OCR box coordinates. We report results on FUNSD and RICO datasets and conclude that `LayerDoc` achieves the SOTA results. Comparing `LayerDoc`'s performance on FUNSD with LayoutLmv1, LayoutLMv2, LayoutReader[46] and ROPE[24] shows competitive p-BLEU performance and reduction in ARD by approximately 10%. For RICO, we compare the reading or-



(a) (b) (c) (d)

Figure 3: Example illustrations of predictions by LayerDoc$_{LLMv2+SBERT}$ on the test set of the Hierarchical Forms dataset. Blue boxes denote input child boxes while green boxes indicate detected parent box in the hierarchy. The pink box in (b) highlights the semantically unique spatial groups inferred from document layout hierarchy.
(a) *Field - widget* pairs are detected with high precision with spatially consistent boxes being grouped together.
(b) Non-trivial form fields aggregated based on their semantic meaning. Eg., addresses in *text fields* are grouped into *text block*.
(c) Extracts difficult non-symmetric *TextBlocks* despite multiple levels of nesting.
(d) Errors in *Choice fields* grouping due to initial mistakes in grouping of *widgets* propagates to later *choice groups* grouping.

der derived from the layout hierarchy as extracted by UIED, Faster-RCNN, DETR, and LayoutLMv2. `LayerDoc` generates better reading order compared to competitive object detection methods. Our contribution becomes significant for RICO dataset where reading order is complicated by deep nested hierarchies. Ablation experiments show that both layout and textual information play equally important roles.

Word Grouping: We observe an improvement of 4% and 10% in F1-score of word grouping performance on FUNSD and RICO datasets, respectively. `LayerDoc` is able to capture the complete text layout that helps it recover missing words that flow over to the start of the next line at line end. This is especially important for grouping check boxes and text fields into choice groups, and table components present in deeply nested scanned forms.

Impact of SBERT and layer-wise structure on `LayerDoc`: We observe a 8-14% performance drop by removing SBERT in element classification, group identification, reading order and word grouping tasks, demonstrating its importance to `LayerDoc` with a LayoutLMv2 backbone. However, even without SBERT, LayerDoc outperforms the LayoutLMv2 baselines on RICO by 10-14%, demonstrating that the layer-wise structure of `LayerDoc` is also important.
Computational cost: On an average, `LayerDoc` requires $\approx$10 times less forward passes to generate complete hierarchy compared to the DocStruct/LayoutLMv2 baseline as it can perform link prediction between a proposed parent box and all child boxes in a layer through its contextual modeling instead of comparing all possible pairs of parent-child pairs across different layers one at a time. This results in reduced search space. LayerDoc has comparable parameters to LayoutLMv2, with the additional parameters from linear layers. Hence their time complexity is comparable, yet LayerDoc outperforms due to algorithmic modifications rather than model size. Figure 3 presents some **illustrative examples** with inferred layout hierarchies by `LayerDoc`.
Error Analysis: (i) Recursive Error Propagation: Figure 4 (appendix) shows that grouping performance reduces higher up the predicted hierarchy as elements detected in the initial layers are used for predicting elements in the subsequent levels of the hierarchy, causing error propagation. (ii) Lack of parent-box context: Our approach infers one parent box at a time in a given layer. Despite optimal layer-wise parent-box selection, errors produced at this step cannot be backpropagated during training. Restricted backtracking in future work may alleviate error accumulation at higher levels.

6. Conclusion and Future Work

We present `LayerDoc` that uses visual, textual and spatial signals along with constraint inference to extract the documents hierarchy in a bottom-up layer-wise fashion. Extensive experiments demonstrate the advantages of our method for extracting specific components of the hierarchy (element type classification and group identification) as well as its downstream applications in reading order detection and word grouping on three diverse semi-structured document datasets. `LayerDoc` enables full-scale hierarchy extraction from diverse documents to enable form authoring, document re-flow, and adaptive editing of user-interfaces. Our current work is limited by its iterative nature and restricted to greedy optimizations. Future work can focus on integrating restricted backtracking in parent selection, layer embedding for different levels, cross-dataset generalization, semi-greedy approaches. Hierarchy construction can aid **long-context document understanding** for tabular parsing [32], layout-enriched speech synthesis [31], and NLP tasks like temporal information extraction [33], temporal dependency parsing [35], and NLI [34].

References

[1] Milan Aggarwal, Hiresh Gupta, Mausoom Sarkar, and Balaji Krishnamurthy. Form2Seq : A framework for higher-order form structure extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3830–3840, Online, Nov. 2020. Association for Computational Linguistics.

[2] Milan Aggarwal, Mausoom Sarkar, Hiresh Gupta, and Balaji Krishnamurthy. Multi-modal association based grouping for form structure extraction. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2064–2073, 2020.

[3] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. *arXiv preprint arXiv:2106.11539*, 2021.

[4] Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornés, and Josep Lladós. Named entity recognition and relation extraction with graph neural networks in semi structured documents. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9622–9627. IEEE, 2021.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[6] Jieshan Chen, Mulong Xie, Zhenchang Xing, Chunyang Chen, Xiwei Xu, Liming Zhu, and Guoqiang Li. Object detection for graphical user interface: old fashioned or deep learning or a combination? *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020.

[7] Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. The significance of reading order in document recognition and its evaluation. In *2013 12th International Conference on Document Analysis and Recognition*, pages 688–692. IEEE, 2013.

[8] Tuan Anh Nguyen Dang, Duc Thanh Hoang, Quang Bach Tran, Chih-Wei Pan, and Thanh Dat Nguyen. End-to-end hierarchical relation extraction for generic form understanding. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5238–5245. IEEE, 2021.

[9] Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, and Curtis Wiginton. Visual fudge: Form understanding via dynamic graph editing. *arXiv preprint arXiv:2105.08194*, 2021.

[10] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, UIST '17, 2017.

[11] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. *ArXiv*, abs/1801.01315, 2018.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[13] Lukasz Garncarek, Rafal Powalski, Tomasz Stanislawek, Bartosz Topolski, Piotr Halama, Michal P. Turski, and Filip Grali'nski. Lambert: Layout-aware language modeling for information extraction. In *ICDAR*, 2021.

[14] Aditya Gupta, Anuj Kumar, Mayank, Vishwa Nath Tripathi, and Sashikala Tapaswi. Mobile web: web manipulation for small displays using multi-level hierarchy page segmentation. In *Mobility '07*, 2007.

[15] Jaekyu Ha, Robert M. Haralick, and Ihsin T. Phillips. Document page decomposition by the bounding-box project. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 2:1119–1122 vol.2, 1995.

[16] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[17] Dafang He, Scott D. Cohen, Brian L. Price, Daniel Kifer, and C. Lee Giles. Multi-scale multi-task fcn for semantic page segmentation and table detection. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:254–261, 2017.

[18] Teakgyu Hong, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model for understanding texts in document. 2020.

[19] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. *arXiv preprint arXiv:2108.04539*, 2021.

[20] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. Spatial dependency parsing for semi-structured document information extraction. *arXiv preprint arXiv:2005.00642*, 2020.

[21] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019.

[22] Mohamed Khemakhem, Axel Herold, and Laurent Romary. Enhancing usability for automatically structuring digitised dictionaries. 2018.

[23] Franck Lebourgeois, Zbigniew Bublinski, and Hubert Emptoz. A fast and efficient method for extracting text paragraphs and graphics from unconstrained documents. *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pages 272–276, 1992.

[24] Chen-Yu Lee, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat, and Tomas Pfister. ROPE: Reading order equivariant positional encoding for graph-based document information extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 314–321, Online, Aug. 2021. Association for Computational Linguistics.

[25] Kai Li, Curtis Wigington, Chris Tensmeyer, Handong Zhao, Nikolaos Barmpalios, Vlad I Morariu, Varun Manjunatha,

Tong Sun, and Yun Fu. Cross-domain document object detection: Benchmark suite and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12915–12924, 2020.

[26] K. Li, Curtis Wigington, Chris Tensmeyer, Handong Zhao, Nikolaos Barmpalios, Vlad I. Morariu, Varun Manjunatha, Tong Sun, and Yun Raymond Fu. Cross-domain document object detection: Benchmark suite and method. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, 2020.

[27] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. DocBank: A benchmark dataset for document layout analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics.

[28] Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multi-modal transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1912–1920, 2021.

[29] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, 2017.

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[31] Puneet Mathur, Franck Dernoncourt, Quan Hung Tran, Jiuxiang Gu, Ani Nenkova, Vlad Morariu, Rajiv Jain, and Dinesh Manocha. Doclayouttts: Dataset and baselines for layout-informed document-level neural speech synthesis. *Proc. Interspeech 2022*, pages 451–455, 2022.

[32] Puneet Mathur, Mihir Goyal, Ramit Sawhney, Ritik Mathur, Jochen Leidner, Franck Dernoncourt, and Dinesh Manocha. Docfin: Multimodal financial prediction and bias mitigation using semi-structured documents. In *Proceedings of the Findings of 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[33] Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. Timers: document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, 2021.

[34] Puneet Mathur, Gautam Kunapuli, Riyaz Ahmad Bhat, Manish Shrivastava, Dinesh Manocha, and Maneesh Singh. Docinfer: Document-level natural language inference using optimal evidence selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[35] Puneet Mathur, Vlad Morariu, Verena Kaynig-Fittkau, Jiuxiang Gu, Franck Dernoncourt, Quan Hung Tran, Ani Nenkova, Dinesh Manocha, and Rajiv Jain. Doctime: A document-level temporal dependency graph parser. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 993–1009, 2022.

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[37] Tom Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. 1999.

[38] Xiaobao Peng, Zhijian Yin, and Zhen Yang. Deeplab_v3_plus-net for image semantic segmentation with channel compression. *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, pages 1320–1324, 2020.

[39] Rafal Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michal Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer. In *ICDAR*, 2021.

[40] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[41] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[42] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[43] Anikó Simon, Jean-Christophe Pret, and A. Peter Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:273–277, 1997.

[44] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[45] Robert Endre Tarjan and Anthony E Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977.

[46] Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. LayoutReader: Pre-training of text and layout for reading order detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4735–4744, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[47] Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. DocStruct: A multimodal method to extract hierarchy structure in document for general form understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 898–908, Online, Nov. 2020. Association for Computational Linguistics.

[48] Zilong Wang, Mingjie Zhan, Houxing Ren, Zhaohui Hou, Yuwei Wu, Xingyan Zhang, and Ding Liang. Grouplink: An end-to-end multitask method for word grouping and relation extraction in form understanding. *ArXiv*, abs/2105.04650, 2021.

[49] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280, June 1989.

[50] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[51] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[52] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei A. F. Florêncio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL/IJCNLP*, 2021.

[53] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C. Lee Giles. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4342–4351, 2017.

[54] Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.

[55] Yue Zhang, Bo Zhang, Rui Wang, Junjie Cao, Chen Li, and Zuyi Bao. Entity relation extraction as dependency parsing in visually rich documents. *arXiv preprint arXiv:2110.09915*, 2021.