Incorporating Visual Layout Structures for Scientific Text Classification

Zejiang Shen¹ Kyle Lo¹ Lucy Lu Wang¹ Bailey Kuehl¹ Daniel S. Weld^{1,2} Doug Downey¹

¹Allen Institute for AI ²University of Washington

{shannons, kylel, lucyw, baileyk, danw, dougd}@allenai.org

Abstract

Classifying the core textual components of a scientific paper-title, author, body text, etc.is a critical first step in automated scientific document understanding. Previous work has shown how using elementary layout information, i.e., each token's 2D position on the page, leads to more accurate classification. We introduce new methods for incorporating VIsual LAyout (VILA) structures, e.g., the grouping of page texts into text lines or text blocks, into language models to further improve performance. We show that the I-VILA approach, which simply adds special tokens denoting the boundaries of layout structures into model inputs, can lead to 1.9% Macro F1 improvements for token classification. Moreover, we design a hierarchical model, H-VILA, that encodes the text based on layout structures and record an up-to 47% inference time reduction with less than 1.5% Macro F1 loss for the text classification models. Experiments are conducted on a newly curated evaluation suite, S2-VLUE, with a novel metric measuring classification uniformity within visual groups and a new dataset of gold annotations covering papers from 19 scientific disciplines. Pre-trained weights, benchmark datasets, and source code will be available at https://github.com/allenai/VILA.

1 Introduction

Scientific papers are usually stored in the Portable Document Format (PDF) without extensive semantic markup. It is critical to extract structured document representations from these PDF files for downstream NLP tasks (Beltagy et al., 2019; Wang et al., 2020) and to improve PDF accessibility (Wang et al., 2021). Unlike other real-world documents, scholarly documents are usually typeset with rectilinear templates of layout structures with interleaving textual and image objects. As shown in Figure 1 (a), while this rich layout information can signal semantics, existing methods (Devlin et al., 2019; Beltagy et al., 2019) resort to analyzing the flattened text obtained by PDF-to-text converters, ignoring the style and layout information stored in the original file.

Recent methods demonstrate that token-level position information, i.e., tokens' 2D spatial location on the page, can be incorporated to improve language models by introducing positionawareness (Xu et al., 2020b) and enhance scientific document parsing (Li et al., 2020). However, humans reading these documents also make use of high-level layout structures like the grouping of text blocks¹ and lines (Todorovic, 2008) to infer the semantic contents accordingly, which we refer to as structure-awareness. Existing models do not explicitly incorporate structure-awareness, resulting in two major drawbacks: 1) deprived of structure-awareness, token-level predictions can be inconsistent within a group (Li et al., 2020); 2) the inference process is less efficient, as redundant predictions need to be made for each token within a group.

In this paper, we investigate to what extent such **VI**sual **LA**yout (VILA) structures can be used to improve NLP models for parsing scholarly documents. Following Zhong et al. (2019) and Tkaczyk et al. (2015)'s work, our key assumption is that a document page can be segmented into visual groups of tokens (either lines or blocks), and the tokens in a group typically have the same semantic category, as shown in Figure 1 (b), which we refer to as the *group uniformity assumption*. We firstly show that VILA can be used as an additional information source to improve existing BERT-based language models (Devlin et al., 2019). Moreover, VILA can also be used as a strong prior that guides the design of model architecture. By injecting spe-

¹In this paper, we consider a text block to be a group of tokens that appear adjacent to each other on a page, are visually set off from other tokens, and belong to the same semantic category. Section headers, lists, equation regions, etc. are valid instances of text blocks.



Figure 1: (a) Real-world scientific documents usually have intricate layout structures, and analyzing the flattened raw text could be sub-optimal. (b) The complex structures could be broken down into groups like text blocks and lines that are composed of tokens with the same semantic category. (c) We design the I-VILA model that injects a special layout indicator token [BLK] into the text inputs to enable structure-awareness. (d) We also develop a hierarchical model, H-VILA, that models each group to reduce the redundant token-level predictions and significantly improve efficiency.

cial layout indicator tokens that delimit boundaries between blocks or lines into existing textual inputs, the **I-VILA** models are *informed* of the document structure, and yield token predictions of better accuracy and consistency. Further, the group uniformity assumption suggests that semantic category prediction can be performed at the group level rather than the token level. We therefore introduce a hierarchical model, **H-VILA**, uses the layout structure to represent a page as a two-level hierarchy of groups and tokens. Because the individual groups can be modeled independently, this reduces the number of tokens that must be fed into the transformer at once, improving efficiency.

The proposed methods are evaluated on a newly designed benchmark suite, the Semantic Scholar Visual Layout-enhanced Scientific Text Understanding Evaluation (S2-VLUE). The benchmark consists of two datasets built on existing resources (Tkaczyk et al., 2015; Li et al., 2020) and a newly curated dataset S2-VL. With high-quality human annotations for papers from 19 disciplines, S2-VL fills important gaps of existing data sets, which only contain machine-generated labels for papers from certain domains. Besides typical measurements of prediction accuracy, we also introduce a novel metric, group category inconsistency, that measures the homogeneity of token classes within each group in terms of entropy. This metric indicates how well the token categories accord with VILA structures.

Our study is well-aligned with recent efforts for

incorporating structural information into language models (Lee et al., 2020; Bai et al., 2020; Yang et al., 2020; Zhang et al., 2019). However, one key difference is that we obtain the structure from layout rather than language structures like sentences or paragraphs. Evaluated on S2-VLUE, we show that when available, VILA structures lead to better prediction accuracy compared to using language structures.

Our main contributions are as follows:

- We explore new ways of injecting layout information in terms of VILA structures into language models, and find that they can improve text classification for scientific text.
- We design two models that incorporate VILA features differently. The I-VILA model injects special layout indicator tokens into the text inputs and improves prediction accuracy (up to +1.9% Macro F1) and consistency compared to previous layout-augmented language models. The H-VILA model performs group-level predictions and can reduce the inference time by 47% with less than 1.5% Macro F1 loss.
- 3. We construct a unified benchmark suite S2-VLUE to systematically evaluate performance on the scientific document text classification task with layout features. We enhance existing datasets with VILA structures, develop a novel dataset S2-VL that addresses gaps in existing resources, with gold labels for 16 token categories and VILA structures for papers from 19 disciplines.

As suggested by improvements on our comprehensive benchmark, our proposed methods also have the potential to benefit PDF extraction in the general domain. The benchmark datasets, modeling code, and trained weights will be available at https://github.com/allenai/VILA.

2 Related Work

The complex organization of scientific paper elements poses a challenge for identifying the key semantic components and converting them to a structured format. Previous work tackles this challenge by utilizing visual or textual features from the documents.

Vision-based Approaches (Zhong et al., 2019; He et al., 2017; Siegel et al., 2018) treat this task as an image object detection problem: given an image of a document page, the models predict a series of rectangular bounding boxes segmenting the page into individual components of different categories. These models excel at capturing complex visual layout structures like figures or tables, but cannot accurately generate fine-grained semantic categories like title, author, or abstract, which are of central importance for parsing scientific documents.

Text-based Methods, on the other hand, formulate this task as a text classification problem. Methods like ScienceParse (Ammar et al., 2018), GROBID (GRO, 2008-2021) or Corpus Conversion Service (Staar et al., 2018) firstly convert a source PDF document to a sequence of tokens via PDF-to-text parsing engines like CER-MINE (Tkaczyk et al., 2015) or pdfalto.² Machine learning models like RNN (Hochreiter and Schmidhuber, 1997), CRF (Lafferty et al., 2001), or Random Forest (Breiman, 2001) trained to model the input texts are then used to classify the token categories. However, without considering document visual structures, these trained models fall short in prediction accuracy or generalize poorly for out-ofdomain documents.

Recently, **Joint Approaches** have been explored that combine visual and textual features to boost model performance. The LayoutLM model (Xu et al., 2020b) combines token textual and 2D position information, and records a 6% F1 improvement over the baseline BERT model (Devlin et al., 2019) on a scientific text classification task (Li et al., 2020). Livathinos et al. (2021) built a

seq2seq model (Sutskever et al., 2014) incorporating both layout and text features that significantly improves model robustness on an evaluation set of diverse paper layouts. Very recent work like LayoutLMv2 (Xu et al., 2020a) and SelfDoc (Li et al., 2021) proposes to incorporate documents' image features when modeling the text, yet it is shown less helpful for text-dense scholarly articles (Li et al., 2021).

The training and evaluation datasets of these models are in many cases automatically generated using paper XML source from PubMed Central (Ammar et al., 2018; GRO, 2008-2021; Tkaczyk et al., 2014) or LaTeX source from arXiv (Li et al., 2020). Despite having large sample sizes, these datasets do not contain significant layout variation, leading to poor generalization to papers from other disciplines with distinct layouts. Also, due to the heuristic nature in which these datasets are constructed, they contain systematic classification errors that can affect downstream modeling performance. We refer the readers to a more detailed description of the limitations for the GROTOAP2 (Tkaczyk et al., 2014) and the DocBank Dataset (Li et al., 2020) in Section 4.1. The PubLayNet dataset (Zhong et al., 2019) provides high-quality text block annotations on 330k document pages; however, its annotations only include five distinct categories, which is insufficient for fully representing the semantic elements found in papers. Livathinos et al. (2021) and Staar et al. (2018) curated a dataset with manual annotations on 2,940 paper pages from various publishers using diverse layouts, but only the processed page features are publicly available, not the raw paper texts or the source PDFs needed for experiments with new layout-aware methods.

3 Method

3.1 **Problem Formulation**

Following the previous literature (Tkaczyk et al., 2015; Li et al., 2020), our task is to map each token t_i from an input document to a semantic category c_i such as title, body text, or reference. For simplicity, we consider only a page of a paper as input; different pages are separately modeled. Input tokens are extracted via PDF-to-text tools, which output both the word w_i and its 2D position, i.e, the rectangular bounding box $a_i = (x_0, y_0, x_1, y_1)$ denoting the left, top, right, and bottom position of the word boundary. Formally, the input sequence

²https://github.com/kermitt2/pdfalto



Figure 2: Visualization of the token level prediction results on the original page. From left to right, we present the ground-truth token category and text block bounding boxes (highlighted in red rectangles), and model predictions from the baseline, I-VILA, and H-VILA model. We use $G^{(B)}$ for VILA-based methods. When VILA is injected, the model achieves more consistent predictions as indicated by arrow (1) and (2) in the figure.

is $T = (t_1, \ldots, t_n)$ where $t_i = \langle w_i, a_i \rangle$ and the output sequence is (c_1, \ldots, c_n) . It's worth noting that the order of tokens in sequence T might not reflect the actual reading order of text due to incorrect PDF-to-text conversion (e.g., in the DocBank dataset (Li et al., 2020)), which is an additional challenge for language models pre-trained on regular texts.

Besides the token sequence T, additional visual structures G can also be retrieved from the source document. Scientific documents are organized into groups of tokens text lines or blocks, which consist of consecutive pieces of text and can be segmented from other pieces based on spatial gaps. The group information can be extracted via visual layout detection models (Zhong et al., 2019; He et al., 2017) or PDF parsing (Tkaczyk et al., 2015).

Formally, given an input page, our system detects a series of m rectangular boxes for each group $B = \{b_1, \ldots, b_m\}$ in the input document page. It further allocates the page tokens to the group region and generates the visual groups $g_i = (b_i, \mathbf{t}_i)$, where $\mathbf{t}_i = \{t_j \mid a_j \leq b_i, t_j \in T\}$ is all tokens in the *i*-th group, and $a_j \leq b_i$ denotes the center point of token t_j 's bounding box a_j is strictly within the group box b_i . When two group regions overlap and share some common tokens, the system will only assign each token to the earlier (in terms of estimated reading order) of the two groups. The token order in each group is consistent with the token order extracted from the page.

We refer to text block groups of a page as $G^{(\mathcal{B})}$ and text line groups as $G^{(\mathcal{L})}$. In our case, we define text lines as consecutive tokens appearing at the nearly same vertical position.³ Text blocks are adjacent text lines with gaps smaller than a certain threshold, and the same semantic category. That is, even two close lines of different semantic categories should be allocated to separate blocks. However, in practice, block or line detectors may generate incorrect predictions.

In the following sections, we show how language models can benefit from these different types of document structures.

3.2 I-VILA: Injecting Visual Layout Indicators

According to the group uniformity assumption, token categories are homogeneous within a group, and categorical changes should happen at group boundaries. This suggests that layout information should be incorporated in a way that informs token *category consistency* intra-group and signals possible token *category changes* inter-group.

Our first method supplies VILA structures via inserting a special layout indicator token at the group boundary in the input text, which we refer to as the I-VILA method. Compared to Xu et al. (2020b), our method provides explicit document structure signals. As shown in Figure 1(c), the inserted tokens highlight individual text segments, resulting in a more structured input for the language models that hints at possible category changes. Additionally, in I-VILA, the special token is seen at all layers of the model, providing VILA signals at different stages of modeling, rather than only providing positional information at the initial em-

³Or horizontal position, when the text is written vertically.

bedding layers (Xu et al., 2020b). We empirically show that BERT-based models can learn to leverage such special tokens to improve both the accuracy and the consistency of category predictions, even without an additional loss penalizing inconsistent intra-group predictions.

In practice, given G, we linearize tokens t_i from each group and flatten them into a 1D sequence T_G . To avoid capturing confounding information in existing pre-training tasks, we insert a new special token [BLK] in-between t_i The resulting input sequence is of the texts. form { [CLS], $\mathbf{t}_{1,1}, \ldots, \mathbf{t}_{i,n_i}$, [BLK], $\mathbf{t}_{i+1,1}, \ldots$, \mathbf{t}_{m,n_m} , [SEP]}, where $\mathbf{t}_{i,j}$ and n_i indicate the *j*th token and the total number of tokens respectively in the *i*-th group, and [CLS] and [SEP] are the special tokens used by the BERT model and are inserted to preserve a similar input structure. The BERT-based models are fine-tuned over the token classification objective with a cross entropy loss. Token positions can also be injected in a similar way as in LayoutLM (Xu et al., 2020b), and the positional embedding for the newly injected [BLK] tokens are derived from the corresponding group's bounding box b_i .

3.3 H-VILA: Visual Layout-guided Hierarchical Model

The uniformity of group token categories also suggests the possibility of building a group-level classifier. Inspired by recent advances in modeling long documents, hierarchical structures (Yang et al., 2020; Zhang et al., 2019) provide an ideal architecture for the end task while optimizing for computational cost. Illustrated in Figure 1(d), two transformer-based models are used to encode each group in terms of its words and modeling the whole document in terms of the groups, respectively, and we provide the modeling details as follows.

The Group Encoder is a l_1 -layer transformer that converts each group g_i into a hidden vector \mathbf{h}_i . Following the typical transformer model setting (Vaswani et al., 2017), the model takes a collection of tokens within a group \mathbf{t}_i as input, and maps each token $\mathbf{t}_{i,j}$ into a dense vector $\mathbf{e}_{i,j}$ of dimension d. Subsequently, a group vector aggregation function $f : \mathbb{R}^{n_i \times d} \to \mathbb{R}^d$ is applied that projects the token representations $(\mathbf{e}_{i,1}, \ldots, \mathbf{e}_{i,n_i})$ to a single vector $\tilde{\mathbf{h}}_i$ that represents the group's textual information. A group's 2D spatial information is incorporated in the form of positional embeddings, and the final group representation h_i can be calculated as:

$$\mathbf{h}_i = \mathbf{h}_i + p(b_i). \tag{1}$$

where p is the 2D positional embedding similar to the one used in LayoutLM:

$$p(b) = E_x(x_0) + E_x(x_1) + E_w(x_1 - x_0) + (2)$$

$$E_y(y_0) + E_y(y_1) + E_h(y_1 - y_0),$$

where E_x , E_x , E_w , E_h are the embedding matrices for x, y coordinates and width and height. We enable model position-awareness at the group level rather than the token level to avoid capturing noisy positional signals from individual tokens. In practice, we find that injecting positional information using the bounding box of the first token within the group leads to better results, and we choose group vector aggregation function f to be the average over all tokens representations.

The Page Encoder is another stacked transformer model of l_2 layers that operates on the group representation h_i generated by the group encoder. Modeling text representations with both structure- and position-awareness, the page encoder efficiently captures layout-contextualized group representations and generates a final group representation s_i optimized for downstream textual analysis. A MLP-based linear classifier is attached thereafter, and is trained to generate the group-level category probability p_{ic} .

Different from previous work (Yang et al., 2020), we restrict the variation in l_1 and l_2 such that we can load pre-trained weights. Therefore, no additional pre-training is required, and the H-VILA model can be fine-tuned directly for the downstream classification task. Specifically, we set $l_1 = 1$ and initialize the group encoder from the first-layer transformer weights of BERT. The page encoder is configured as either a one-layer transformer or a 12layer transformer that resembles a full LayoutLM model. Weights are initialized from the first-layer or full 12 layers of the LayoutLM model, which is trained to model texts in conjunction with their positions.

Group Token Truncation As suggested in Yang et al. (2020)'s work, when an input document of length N is evenly split into segments of L_s , the memory footprint of the hierarchical model is $O(l_1NL_s + l_2(\frac{N}{L_s})^2)$, and for long documents with $N \gg L_d$, it approximates as $O(N^2/L_s^2)$. However,

	GROTOAP2	DocBank	S2-VL
Train / Dev / Test Pages	834k / 18k / 18k	398k / 50k / 50k	1.3k ¹
Annotation Method	Semi-Automatic	Automatic	Human Annotation
Paper Domain	Life Science	Math / Physics / CS	19 Domains
VILA Structure	PDF parsing	Vision model	Gold Label / Detection methods
# of Categories	22	12	16
# Tokens Per Page	1203 / 591 / 2307 ²	838 / 503/ 1553	790 / 453 / 1591
# Tokens Per Text Block	90 / 184 / 431	57 / 138 / 210	48 / 121 / 249
# Tokens Per Text Line	17 / 12 / 38	16 / 43 / 38	14 / 10 / 30
# Text Lines Per Page	90/51/171	60 / 34 / 125	64 / 54 / 154
# Text Blocks Per Page	12/16/37	15 / 8 / 30	22 / 36 / 68

¹ This is the total number of pages in the S2-VL dataset; we use 5-fold cross-validation for training and testing. ² For this and all following cells, we report the average / standard deviation / 95-th percentile value for this item.

Table 1: Distinct features for the three datasets in the S2-VLUE benchmark.

in our case, it is infeasible to adopt the Greedy Sentence Filling technique (Yang et al., 2020) as it will mingle signals from different groups and obfuscate group structures. It's also less desirable to simply use the maximum token count per group $\max_{1 \le i \le m} n_i$ to batch the contents due to the high variance of group token length (see Table 1). Instead, we choose a group token truncation count \tilde{n} empirically based on key stats of the group token length distribution such that $N \approx \tilde{n}m$, and use the first \tilde{n} to aggregate the group hidden vector \mathbf{h}_i for all groups. Therefore, H-VILA models can achieve similar efficiency gains as seen in the previous method (Yang et al., 2020).

4 Benchmark Suite

To systematically evaluate the proposed methods, we develop the the Semantic Scholar Visual Layout-enhanced Scientific Text Understanding Evaluation (S2-VLUE) benchmark suite. S2-VLUE consists of three datasets—two previously released resources augmented with VILA information and a newly curated dataset S2-VL—as well as evaluation metrics for measuring prediction quality.

4.1 Datasets

Key statistics for S2-VLUE are provided in Table 1. Notably, the three constituent datasets differ in regards to: 1) annotation method, 2) VILA generation method, and 3) paper domain coverage. We provide details below.

GROTOAP2 The GROTOAP2 dataset (Tkaczyk et al., 2014) is semi-automatically annotated. It first creates text block and line groupings using the CERMINE PDF parsing tool (Tkaczyk et al.,

2015); text block category labels are then obtained by pairing block texts with structured data from document source files obtained from PubMed Central. A small subset of data is inspected by experts, and a set of post-processing heuristics is developed to further improve annotation quality. Since token categories are annotated by group, the dataset achieves perfect accordance between token labels and VILA structure. However, the method of rulebased PDF parsing employed by the authors introduces labeling inaccuracies due to imperfect VILA detection: the authors find that block-level annotation accuracy achieves only 92 Macro F1 in a small gold evaluation set. Additionally, all samples are extracted from the PMC Open Access Subset⁴ that includes only life sciences publications; these papers have less representation of classification types like "equation", which are common in other scientific disciplines.

DocBank The DocBank dataset (Li et al., 2020) is fully machine-labeled without any postprocessing heuristics or human assessment. The authors first identify token categories by automatically parsing the source TEX files available from arXiv. Text block annotations are then generated by grouping together tokens of the same category using connected component analysis. However, only a specific set of token tags was extracted from the main TEX file for each paper, leading to inaccurate and incomplete token labels, especially for papers employing LaTeX macro commands,⁵ and thus in-

⁴https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

⁵For example, in DocBank, "Figure 1" in a figure caption block is usually labeled as "paragraph" rather than "caption". DocBank labels all tokens that are not explicitly contained in the set of processed LaTeX tags as "paragraph."

correct visual groupings. Hence, we develop a CNN-based vision layout detection model based on a collection of existing resources (Zhong et al., 2019; MFD, 2021; He et al., 2017; Shen et al., 2021) to fix these inaccuracies and generate trustworthy VILA annotations at both the text block and line level.⁶ As a result, this dataset can be used to evaluate VILA models under a different setting, since the VILA structures are generated independently from the token annotations. Yet as the papers are extracted from arXiv, they primarily represent domains like Computer Science, Physics, and Mathematics, limiting the amount of layout variation; the automatic token classification process also produces labels for only a small number of semantic categories, which may be insufficient for some paper modeling needs.

S2-VL S2-VL is created to address the three major drawbacks in existing work: 1) annotation quality, 2) VILA creation, and 3) domain coverage. S2-VL is developed with help from graduate students who frequently read scientific papers. Using the PAWLS annotation tool (Neumann et al., 2021), annotators draw rectangular text blocks directly on each PDF page, and specify the block-level semantic categories from 16 possible candidates. Tokens within a group can therefore inherit the category from the parent text block. Inter-annotator agreement, in terms of token-level accuracy measured on a 12-paper subset, is high at 0.95. The ground-truth VILA labels in S2-VL can be used to fine-tune visual layout detection models, and paper PDFs are also included, making PDF-based structure parsing feasible: this enables VILA annotations to be created by different means, which is helpful for benchmarking VILA-based models in different scenarios. Moreover, S2-VL currently contains 1337 pages of 87 papers from 19 different disciplines, including Philosophy and Sociology that are not present in previous data sets.

Overall, the datasets in S2-VLUE cover a wide range of scientific disciplines with different layouts. The VILA structures are curated differently, and can be used to evaluate a variety of VILA-based methods and assess their generalizability.

4.2 Metrics

Prediction Accuracy The token label distribution is heavily skewed towards categories indicating papers' body texts (e.g., the "BODY_CONTENT" category in GROTOAP2 or the "paragraph" category in S2-VL and DocBank). Therefore, we choose to use Macro F1 as the main evaluation metric for prediction accuracy.

Group Category Inconsistency We also develop a metric that calculates the uniformity of the token categories within a group. Hypothetically, tokens in $T_g^{(i)}$ share the same category c, and naturally the group inherits the semantic label c. We use the group token category entropy to measure the inconsistency of the (predicted) token categories within a group:

$$H(g) = -\sum_{c} p_c \log p_c, \qquad (3)$$

where p_c denotes the probability of a token in group g being classified as category c. When all tokens in a group have the same category, the group token category inconsistency is zero. H(g) reaches the maximum when p_c is a uniform distribution across all possible categories. The inconsistency for G is the arithmetic mean of all individual groups g_i :

$$\mathbf{H}(G) = \frac{1}{m} \sum_{i}^{m} \mathbf{H}(g_i) \tag{4}$$

H(G) acts as an auxiliary metric for evaluating prediction quality with respect to the provided VILA structures. In the remainder of this paper, we report the inconsistency metric for text blocks $G^{(B)}$ and scale the values up by a factor of 100.

5 Experiments

5.1 Experimental Setup

Implementation Details Our models are implemented using PyTorch (Paszke et al., 2019) and the transformers library (Wolf et al., 2020). A series of baseline and VILA models are fine-tuned on 4-GPU RTX8000 or A100 machines. The AdamW optimizer (Kingma and Ba, 2014; Loshchilov and Hutter, 2019) is adopted with a 5×10^{-5} learning rate and (β_1 , β_2) = (0.9, 0.999). The learning rate is linearly warmed up over 5% steps then linearly decayed. For different datasets (GROTOAP2, DocBank, S2-VL), unless otherwise specified, we select the best fine-tuning batch size (40, 40 and 12)

⁶The original generation method for DocBank requires rendering LaTeX source, which results in layouts different from the publicly available versions of these documents on arXiv. However, because the authors of the dataset only provide document page images, rather than the rendered PDF, we can only use image-based approaches for layout detection. We refer readers to supplementary materials for details.

		GROTO	AP2	DocBa	unk	S2-VL		
		F1-macro †↑	$\mathrm{H}(G) \amalg$	F1-macro ↑↑	$\mathrm{H}(G) {\downarrow \!\!\!\downarrow}$	F1-macro ↑↑	$\mathrm{H}(G) {\downarrow \!\!\!\downarrow}$	Inference Time (ms)
Baseline	LayoutLM	92.34	0.78	91.06	3.04	81.65(5.05) ¹	3.56(0.62)	52.56(0.25)
	Sentence	91.83	0.78	91.44	3.03	82.03(4.20)	3.50(0.38)	54.09(0.37)
I-VILA	Text Line $G^{(\mathcal{L})}$	92.37	0.73	92.79	2.59	82.87(3.95) ²	3.22(0.53)	56.31(0.40)
	Text Block $G^{(\mathcal{B})}$	93.38	0.53	92.00	2.51	82.65(4.90)	2.51(0.44)	53.27(0.13)
	Naïve	92.65	0.00	87.01	0.00	75.60(4.72)	0.38(0.17)	82.57(0.30)
H-VILA	Text Line $G^{(\mathcal{L})}$	91.65	0.32	91.27	1.07	80.47(2.03)	2.62(0.70)	28.07(0.37)
	Text Block $G^{(\mathcal{B})}$	92.37	0.00	87.78	0.00	76.06(4.66)	0.38(0.22)	16.37(0.15)

¹ For S2-VL, we show the averaged scores and their standard deviation in the parentheses across the 5-fold cross validation subsets.

 2 In this table, we report S2-VL results using VILA structures detected by visual layout models. When the ground-truth VILA structures are available, both I-VILA and H-VILA models can achieve better accuracy, shown in Table 5 and 6.

³ When reporting efficiency in other parts of the paper, we use this result because of its optimal combination of accuracy and efficiency.

Table 2: Model Performance on the Three Benchmark Datasets. Models with layout indicator achieves better accuracy, while hierarchical models achieves significant efficiency gain compared to the baseline methods.

and training epochs (24, 6,⁷ and 10) for all models. As for S2-VL, given its smaller size, we use 5-fold cross validation and report averaged scores, and use 2×10^{-5} learning rate with 20 epochs. Since papers may have variable page numbers, we split S2-VL based on papers rather than pages to avoid exposing paper templates of test samples in the training data. Mixed precision training (Micikevicius et al., 2018) is used to speed up the training process.

For I-VILA models, we fine-tune several BERTvariants with VILA-enhanced text inputs, and the models are initialized from pre-trained weights available in the transformers library. The H-VILA models are initialized as mentioned in Section 3.3, and by default, positional information is injected for each group.

Competing Methods We consider three approaches that compete with the proposed methods from different perspectives: 1) The LayoutLM (Xu et al., 2020b) model is the baseline method. It is the closest model counterpart to our VILA-augmented models as it also injects layout information, and achieves previous SOTA performance on the Scientific PDF parsing task (Li et al., 2020). 2) For indicator-based methods, besides using VILA-based indicators, we also compare with indicators generated from *sentence* breaks detected by PySBD (Sadvilkar and Neumann, 2020). 3) For hierarchical models, we consider a *naïve* approach

where the group texts are separately fed into a LayoutLM-based group token classifier. However, despite the group texts being relatively short, this method causes extra computational overhead as the full LayoutLM model needs to be run $m \gg 3$ times for all groups.⁸ As such, we only consider text block groupings in this case for efficiency, and models are only trained for 5, 2, and 5 epochs for GROTOAP2, DocBank, and S2-VL.

Measuring Efficiency We report the inference time per sample as a measure of model efficiency. We select 1,000 pages from the GROTOAP2 test set, and report the average model runtime for 3 runs on this subset. All models are tested on an isolated machine with a single V100 GPU. We report the time incurred for text classification; additional time costs associated with PDF-to-text conversion or VILA structure detection are not included.

5.2 Evaluation Results

Summarized in Table 2, VILA-based approaches achieve better accuracy or efficiency under different scenarios.

I-VILA models lead to consistent accuracy improvements when layout information is injected. Compared to the baseline LayoutLM model, which uses regular textual input, inserting layout indicators results in +1.1%, +1.9%, and +1.5% Macro F1 improvements across the three benchmark datasets. I-VILA models also achieve better token prediction

⁷We try to keep gradient update steps the same for the GROTOAP2 and the DocBank dataset. As DocBank contains $4 \times$ examples, the number of DocBank models' training epochs is reduced by 75%.

⁸Using GROTOAP2 as an example (see Table 1), there is an average of 12 text blocks per page and $m \approx 12$. With an average of 1203 tokens per page, the inputs must be split into three 512-length segments that are separately provided as inputs into the LayoutLM model.

Base Model	I-VILA	F1-marco ↑↑	$\mathrm{H}(G) \amalg$
	None	90.52	1.95
DistilBERT	Text Line	91.14	1.33
	Text Block	92.12	0.73
BERT	None	90.78	1.58
	Text Line	91.65	1.13
	Text Block	92.31	0.63
	None	92.34	0.78
LayoutLM	Text Line	92.37	0.73
	Text Block	93.38	0.53

Table 3: Inserting VILA indicator tokens leads to consistent improvements on different BERT-based models.

consistency; the corresponding group category inconsistency is reduced by 32.1%, 14.8%, and 9.6% compared to baseline. Moreover, VILA information is also more helpful than language structures: I-VILA models based on text blocks and lines all outperform the sentence boundary-based method by a similar margin. Figure 2 shows an example of the VILA model predictions.

H-VILA models, on the other hand, lead to significant efficiency improvements. In Table 2, we report results for H-VILA models with $l_1 = 1$ and $l_2 = 12$. As block-level models perform prediction directly at the text block level, the group category inconsistency is naturally zero. Compared to LayoutLM, H-VILA models with text lines brings 46.59% reduction in the inference time, while the final prediction accuracy is not heavily penalized (-0.74%, +0.23%, -1.45% Macro F1). When text blocks are used, H-VILA models are even more efficient (68.85% and 80.17% inference time reduction compared to the LayoutLM and naïve baseline), and they also achieves better accuracy compared to the naïve counterparts (-0.30%, 0.88%, 0.61%) Macro F1).

However, in H-VILA models, the inductive bias from the group uniformity assumption is a doubleedged sword: models are often less accurate than I-VILA counterparts, and performing block level classification may sometimes lead to worse results (-3.6% and -6.8% Macro F1 in the DocBank and S2-VL datasets compared to LayoutLM). Shown in Figure 3, when text block detection is incorrect, the H-VILA method lacks the flexibility to assign different token categories within a group, which can lead to lower accuracy. Text lines contain shorter

 $G^{(b)}$ Source: Rule-based PDF Parsing



Figure 3: Different from Figure 2, here we show models trained and evaluated with "inconsistent" text block detections. The blocks are created by the CERMINE PDF parsing program (Tkaczyk et al., 2015), which (1) fails to capture the correct table structure and (2) does not separate body text contents into different blocks. Though VILA-based models utilize the group structure to increase the block uniformity, overall prediction accuracy is hurt due to the inconsistent block signal.

token sequences than blocks, and the empirical risk of erroneous predictions are consequently lower. Additional analysis of the two different H-VILA models and VILA signals is detailed in Section 6.

5.3 I-VILA on different BERT variants

We also apply I-VILA to different BERT variants on the GROTOAP2 dataset, and show that the method leads to consistent improvements for all model variants in Table 3. For BERT and DistilBERT (Sanh et al., 2019), 2D coordinates for layout indicator tokens are not injected. Yet we still observe consistent improvements (+1.68% and +1.76% Macro F1) compared to non-VILA counterparts, showing that VILA structural information can be used separately from positional information. Moreover, I-VILA + BERT has nearly identical performance as LayoutLM. This further verifies that injecting layout indicator tokens is a novel and effective way of incorporating layout information into language models.

5.4 Ablating the Optimal Configurations for H-VILA

Next we analyze different architectures of the H-VILA models using the GROTOAP2 dataset. By varying the transformer layers l_2 in the page en-

		Text Line			Text Block		
l_2	Use 2D Position	F1-macro	$\mathrm{H}(G)$	Inference Time	F1 macro	$\mathrm{H}(G)$	Inference Time
1	X	89.77	1.73	24.23(0.06)	91.93	0.00	15.90(0.19)
1	1	91.30	1.17	24.47(0.45)	91.89	0.00	16.05(0.28)
12	X	91.07	0.63	27.60(0.11)	92.14	0.00	16.65(0.09)
12	1	91.65	0.32	28.07(0.37)	92.37*	0.00	16.37(0.15)

Table 4: Model performances for different H-VILA architectures.

coder, we aim to find the best configurations with an optimal balance between accuracy and efficiency. We also investigate the importance of 2D group positions by experimenting with only using the textual representation $\tilde{\mathbf{h}}_i$ in the page encoders. Results are presented in Table 4.

As mentioned previously, we experiment with $l_2 \in \{1, 12\}$ in order to take advantage of existing pre-trained BERT weights. Interestingly, even with only a two layer structure (when $l_2 = 1$), the model can capture sufficient semantic information, and generate predictions with good accuracy. This formulation brings considerable efficiency improvements, with the most accurate model (marked with \star in the table) beating the baseline LayoutLM by 68% in terms of efficiency, let alone the naïve baseline (80%). This model also outperforms the baseline DistilBERT model with 90.52 Macro F1 and 26.48 ms Inference Time.

We observe consistent improvements when 2D positional information is injected into the group representations. Models are able to leverage both structure and positional information to generate predictions of better accuracy and consistency: when text lines are used, modeling with groups' 2D positions leads to lower group category inconsistency.

6 Discussion

Though we treat text lines and blocks (almost) interchangeably in the previous sections, using $G^{(\mathcal{B})}$ and $G^{(\mathcal{L})}$ may lead to different modeling outcomes. In this section, we try to answer questions about 1) which grouping level is preferred and 2) what is the best way to obtain these groupings. We start with a careful analysis of the definition of the text "blocks" and "lines", and identify possible issues in text blocks that might violate the group token uniformity assumption. Additional experiments are implemented where we compare different group detection methods on the S2-VL dataset, and draw conclusions that might be helpful for practical use.

	I-VILA			H-VILA		
$G^{(\mathcal{B})}$ Source	F1-macro	$\mathrm{H}(G)$		F1-macro	$\mathrm{H}(G)$	
Ground-Truth	86.09(4.76)	1.90(0.46)		79.32(3.65)	0.45(0.28)	
PDF Parsing	81.88(5.18)	3.67(0.87)		75.17(4.10)	0.02(0.01)	
Vision Model	82.65(4.90)	2.51(0.44)		76.06(4.66)	0.38(0.22)	

Table 5: Model performances when using different $G^{(\mathcal{B})}$ for training and evaluation on the S2-VL dataset.

	I-VILA			H-VILA		
$G^{(\mathcal{L})}$ Source	F1-macro	$\mathrm{H}(G)$	-	F1-macro	$\mathrm{H}(G)$	
PDF Parsing	82.40(4.58)	3.38(0.77)		78.88(3.89)	2.52(0.62)	
Vision Model	82.87(3.95)	3.22(0.53)		80.47(2.03)	2.62(0.70)	

Table 6: Model performances when using different $G^{(\mathcal{L})}$ for training and evaluation on the S2-VL dataset.

6.1 Text Blocks vs. Text Line

Text lines are well-studied in layout analysis, partly because of the relatively unambiguous definition: consecutive texts appearing at the same vertical position⁹ are considered as a text line. Text blocks, however, are less well-defined and the may vary in different contexts. For example, section headers and paragraphs are labeled in separate text blocks in some cases (Zhong et al., 2019)), while they are merged as a single text block in other cases due to their spatial proximity on the page (Tkaczyk et al., 2015). If block detection is inconsistent with the token semantic schema (e.g., when the model is required to differentiate section headers and paragraph texts yet they are included in the same text block), the block signal might be less helpful or even hurt VILA-based models.

6.2 S2-VL with Different Group Detectors

To verify this idea, we investigate different $G^{(B)}$ derived via PDF parsing (Tkaczyk et al., 2015) or vision model detection (He et al., 2017) for the S2-VL dataset. We train and evaluate our models on

⁹or horizontal position when the text are written vertically.



Figure 4: Few-shot learning experiments on the S2-VL dataset with the ground-truth text block annotations.

these data variants, and report the performance in Table 5. As shown in Figure 3, the PDF parsing engine's block detection is inconsistent with our token labeling schema, and the detected blocks usually contain tokens of different categories. I-VILA models record less accuracy improvements in this case, and H-VILA methods are penalized heavily as they, by design, can only generate the same category prediction for all tokens in the same group.

We also vary the text line detection methods, retrain the models, and report performance in Table 6. Similarly, the vision model yields better line detection results, and thus leads to better performance in the VILA-based models.

We conclude with some practical suggestions for using VILA-based methods.

- 1. When consistent text blocks are available, $G^{(\mathcal{B})}$ is preferred as it can lead to better accuracy and efficiency gains. Equivalently, when creating new datasets, it is ideal to design the token labeling schema to align with the block detection method.
- 2. Otherwise, using $G^{(\mathcal{L})}$ or I-VILA is a viable option that can lead to consistent modeling improvements.
- 3. Despite being more computationally expensive, using a vision detection model can lead to more robust accuracy improvements.

6.3 Few-shot Experiments on S2-VL

Finally, we show that I-VILA methods can help with model training and improve sample efficiency when the perfect text block information is given. We conduct few-shot learning experiments on the S2-VL dataset, training models using samples from 5, 10, 15, 25, and 45 papers. Similar to previous settings, 5-fold cross validation is applied to each few-shot experiment, and train-test splits are held constant while varying training sample size. Equipped with text block indicators, I-VILA models trained on a 15-paper subset can outperform the baseline LayoutLM model trained on the full dataset of 70 papers. However, the text line based I-VILA models lead to relatively less improvements compared to the text block based counterpart. This difference could be explained by the distinct text block and line count per page (Table 1). With 50% more text lines on a page, the special token is injected into the inputs more frequently. This can cause a bigger difference in the text structure than what the models are pre-trained on, and the frequent appearance of such tokens may diminish their relative importance.

7 Conclusion

In this paper, we introduce two new ways to integrate Visual Layout (VILA) structures into the NLP pipeline for analyzing scientific documents. We show that inserting special indicator tokens based on VILA (I-VILA) can lead to robust improvements in token classification accuracy (up to +1.9% Macro F1) and consistency (up to -32% group category inconsistency). In addition, we design a hierarchical transformer model based on VILA (H-VILA), which can reduce inference time by 46% with less than 1.5% Macro F1 reduction compared to previous SOTA methods. We release a benchmark suite, along with a newly curated dataset S2-VL, to systematically evaluate the proposed methods. We ablate the influence of different visual layout detectors on VILA-based models, and provide suggestions for practical use. Our study is well-aligned with the recent exploration of injecting structure into language models, and provides new perspectives on how to incorporate visual structures.

References

- 2008-2021. Grobid. https://github.com/ kermitt2/grobid.
- 2021. Icdar2021 competition on mathematical formula detection. http://transcriptorium. eu/~htrcontest/MathsICDAR2021/. Accessed: 2021-04-30.
- Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Ja-

son Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 84–91, New Orleans - Louisiana. Association for Computational Linguistics.

- He Bai, Peng Shi, Jimmy Lin, Yuqing Xie, Luchen Tan, Kun Xiong, Wen Gao, and Ming Li. 2020. Segatron: Segment-aware transformer for language modeling and understanding. arXiv preprint arXiv:2004.14996.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciB-ERT: A pretrained language model for scientific text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3615– 3620, Hong Kong, China. Association for Computational Linguistics.
- Leo Breiman. 2001. Random forests. *Machine learn-ing*, 45(1):5–32.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Haejun Lee, Drew A. Hudson, Kangwook Lee, and Christopher D. Manning. 2020. SLM: Learning a discourse language representation with sentence unshuffling. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1551–1562, Online. Association for Computational Linguistics.

- Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis.
- Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. 2021. Selfdoc: Self-supervised document representation learning. arXiv preprint arXiv:2106.03331.
- Nikolaos Livathinos, Cesar Berrospi, Maksym Lysak, Viktor Kuropiatnyk, Ahmed Nassar, Andre Carvalho, Michele Dolfi, Christoph Auer, Kasper Dinkla, and Peter Staar. 2021. Robust pdf document conversion using recurrent neural networks. *arXiv preprint arXiv:2102.09395*.
- I. Loshchilov and F. Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.
- P. Micikevicius, Sharan Narang, Jonah Alben, G. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, O. Kuchaiev, Ganesh Venkatesh, and H. Wu. 2018. Mixed precision training. *ArXiv*, abs/1710.03740.
- Mark Neumann, Zejiang Shen, and Sam Skjonsberg. 2021. Pawls: Pdf annotation with labels and structure. *arXiv preprint arXiv:2101.10281*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.
- Nipun Sadvilkar and Mark Neumann. 2020. PySBD: Pragmatic sentence boundary disambiguation. In Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), pages 110–114, Online. Association for Computational Linguistics.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv*:1910.01108.
- Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. Layoutparser: A unified toolkit for deep learning based document image analysis. *arXiv preprint arXiv:2103.15348*.
- Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*, pages 223–232.

- Peter WJ Staar, Michele Dolfi, Christoph Auer, and Costas Bekas. 2018. Corpus conversion service: A machine learning platform to ingest documents at scale. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 774–782.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Dominika Tkaczyk, Pawel Szostek, and Lukasz Bolikowski. 2014. Grotoap2—the methodology of creating a large ground truth dataset of scientific articles. *D-Lib Magazine*, 20(11/12).
- Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJ-DAR)*, 18(4):317–335.
- Dejan Todorovic. 2008. Gestalt principles. *Scholarpedia*, 3(12):5345.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Lucy Lu Wang, Isabel Cachola, Jonathan Bragg, Evie Yu-Yen Cheng, Chelsea Haupt, Matt Latzke, Bailey Kuehl, Madeleine van Zuylen, Linda Wagner, and Daniel S Weld. 2021. Improving the accessibility of scientific documents: Current state, user needs, and a system solution to enhance scientific pdf accessibility for blind and low vision users. *arXiv e-prints*, pages arXiv–2105.
- Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. 2020. Cord-19: The covid-19 open research dataset. *ArXiv*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. arXiv preprint arXiv:2012.14740.

- Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. Layoutlm: Pretraining 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.
- Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pages 1725–1734.
- Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HI-BERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.
- Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1015–1022. IEEE.