VILA: Improving Structured Content Extraction from Scientific PDFs Using Visual Layout Groups

Zejiang Shen[1] **Kyle Lo**[1] **Lucy Lu Wang**[1] **Bailey Kuehl**[1]
Daniel S. Weld[1,2] **Doug Downey**[1,3]
[1]Allen Institute for AI [2]University of Washington [3]Northwestern University
{shannons, kylel, lucyw, baileyk, danw, dougd}@allenai.org

Abstract

Accurately extracting structured content from PDFs is a critical first step for NLP over scientific papers. Recent work has improved extraction accuracy by incorporating elementary layout information, e.g., each token's 2D position on the page, into language model pretraining. We introduce new methods that explicitly model VIsual LAyout (VILA) *groups*, i.e., text lines or text blocks, to further improve performance. In our I-VILA approach, we show that simply inserting special tokens denoting layout group boundaries into model inputs can lead to a 1.9% Macro F1 improvement in token classification. In the H-VILA approach, we show that hierarchical encoding of layout-groups can result in up-to 47% inference time reduction with less than 0.8% Macro F1 loss. Unlike prior layout-aware approaches, our methods do not require expensive additional pretraining, only fine-tuning, which we show can reduce training cost by up to 95%. Experiments are conducted on a newly curated evaluation suite, S2-VLUE, that unifies existing automatically-labeled datasets and includes a new dataset of manual annotations covering diverse papers from 19 scientific disciplines. Pre-trained weights, benchmark datasets, and source code are available at https://github.com/allenai/VILA.

1 Introduction

Scientific papers are usually distributed in Portable Document Format (PDF) without extensive semantic markup. Extracting structured document representations from these PDF files—i.e., identifying title and author blocks, figures, references, and so on—is a critical first step for downstream NLP tasks (Beltagy et al., 2019; Wang et al., 2020) and is important for improving PDF accessibility (Wang et al., 2021).

Recent work demonstrates that document layout information can be used to enhance content extraction via large-scale, layout-aware pretraining (Xu et al., 2020, 2021; Li et al., 2021). However, these methods only consider individual tokens' 2D positions and do not explicitly model high-level layout structures like the grouping of text into lines and blocks (see Figure 1 for example), limiting accuracy. Further, existing methods come with enormous computational costs: they rely on further pretraining an existing pretrained model like BERT (Devlin et al., 2019) on layout-enriched input, and achieving the best performance from the models requires more than a thousand (Xu et al., 2020) to several thousand (Xu et al., 2021) GPU-hours. This means swapping in a new pretrained text model or experimenting with new layout-aware architectures can be prohibitively expensive, incompatible with the goals of green AI (Schwartz et al., 2020).

In this paper, we explore how to improve the accuracy and efficiency of structured content extraction from scientific documents by using **VI**sual **LA**yout (VILA) groups. Following Zhong et al. (2019) and Tkaczyk et al. (2015), our methods use the idea that a document page can be segmented into visual groups of tokens (either lines or blocks), and that the tokens within each group generally have the same semantic category, which we refer to as the *group uniformity assumption* (see Figure 1(b)). Given text lines or blocks generated by rule-based PDF parsers (Tkaczyk et al., 2015) or vision models (Zhong et al., 2019), we design two different methods to incorporate the VILA groups and the assumption into modeling: the **I-VILA** model adds layout **i**ndicator tokens to textual inputs to improve the accuracy of existing BERT-based language models, while the **H-VILA** model uses VILA structures to define a **h**ierarchical model that models pages as collections of groups rather than of individual tokens, increasing inference efficiency.

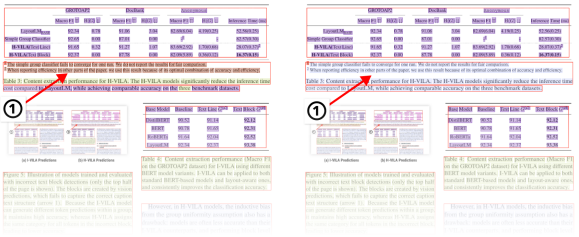
	GROTOAP2		DocBank		S2-VL		
	Macro F1 ↑↑	H(G) ↓↓	Macro F1 ↑↑	H(G) ↓↓	Macro F1 ↑↑	H(G) ↓↓	Inference Time (ms)
$\text{LayoutLM}_{\text{BASE}}$	92.34	0.78	91.06	2.64	82.69(6.04)	4.19(0.25)	52.56(0.25)
Simple Group Classifier	92.65	0.00	87.01	0.00	-[1]	-	82.57(0.30)
H-VILA(Text Line)	91.65	0.32	91.27	1.07	83.69(2.92)	1.70(0.68)	28.07(0.37)[2]
H-VILA(Text Block)	92.37	0.00	87.78	0.00	82.09(5.89)	0.36(0.12)	**16.37(0.15)**

[1] The simple group classifier fails to converge for one run. We do not report the results for fair comparison.
[2] When reporting efficiency in other parts of the paper, we use this result because of its optimal combination of accuracy and efficiency.

Table 3: Content extraction performance for H-VILA. The H-VILA models significantly reduce the inference time cost compared to LayoutLM, while achieving comparable accuracy on the three benchmark datasets.

Base Model	Baseline	Text Line $G^{(\mathcal{L})}$	Text Block $G^{(\mathcal{B})}$
DistilBERT	90.52	91.14	**92.12**
BERT	90.78	91.65	**92.31**
RoBERTa	91.64	92.04	**92.52**
LayoutLM	92.34	92.37	**93.38**

Table 4: Content extraction performance (Macro F1 on the GROTOAP2 dataset) for I-VILA using different BERT model variants. I-VILA can be applied to both standard BERT-based models and layout-aware ones, and consistently improves the classification accuracy.



(a) I-VILA Predictions (b) H-VILA Predictions

Figure 5: Illustration of models trained and evaluated with incorrect text block detections (only the top half of the page is shown). The blocks are created by vision predictions, which fails to capture the correct caption text structure (arrow 1). Because the I-VILA model can generate different token predictions within a group, it maintains high accuracy, whereas H-VILA assigns the same category for all tokens in the incorrect block, leading to lower accuracy.

6 Results

6.1 I-VILA Achieves Better Accuracy

Table 2 shows that I-VILA models lead to consistent accuracy improvements without further pretraining. Compared to the baseline LayoutLM model, inserting layout indicators results in +1.13%, +1.90%, and +1.29% Macro F1 improvements across the three benchmark datasets. I-VILA models also achieve better token prediction consistency; the corresponding group category inconsistency is reduced by 32.1%, 21.7%, and 21.7% 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 4 shows an example of the VILA model predictions.

6.2 H-VILA is More Efficient

Table 3 summarizes the efficiency improvements of the H-VILA models with $l_{\text{g}} = 1$ and $l_{\text{p}} = 12$. As block-level models perform predictions directly at the text block level, the group category inconsistency is naturally zero. Compared to LayoutLM, H-VILA models with text lines brings a 46.59% reduction in inference time, without heavily penalizing the final prediction accuracies (-0.75%, +0.23%, +1.21% 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 simple group classifier baseline), and they also achieve similar or better accuracy compared to the simple group classifier (-0.30%, +0.88% Macro F1 for GROTOAP2 and DocBank).

However, in H-VILA models, the inductive bias from the group uniformity assumption also has a drawback: models are often less accurate than their I-VILA counterparts, and performing block level classification may sometimes lead to worse results (-3.60% and -0.73% Macro F1 in the DocBank and S2-VL datasets compared to LayoutLM). Moreover, shown in Figure 5, when the injected layout group is incorrect, the H-VILA method lacks the flexibility to assign different token categories within a group, leading to lower accuracy. Additional analysis of the impact of the layout group predictions is detailed in Section 8.

	Abstract	Acknowledgment	Affiliation	Author	Author Title	Bib Info	Body Content	Conflict Statement
$BERT_{BASE}$	97.42	95.83	96.12	96.91	96.09	95.00	98.80	88.66
$BERT_{BASE}$ + **I-VILA**(Text Line)	97.65	95.89	96.61	97.17	96.48	95.78	98.93	88.28
$BERT_{BASE}$ + **I-VILA**(Text Block)	97.67	96.46	96.80	97.23	97.73	96.29	98.99	91.88
$LayoutLM_{BASE}$	98.05	96.29	96.64	**97.49**	96.51	96.74	99.06	91.16
$LayoutLM_{BASE}$ + Sentence Breaks	97.92	96.32	96.68	96.74	95.42	96.77	99.11	90.42
$LayoutLM_{BASE}$ + **I-VILA**(Text Line)	97.99	96.41	96.72	97.29	95.98	96.66	99.11	90.75
$LayoutLM_{BASE}$ + **I-VILA**(Text Block)	98.12	**96.81**	96.93	96.96	97.52	**96.87**	99.14	91.43
Simple Group Classifier	96.10	95.53	**97.10**	97.48	**97.94**	96.68	98.94	**93.25**
H-VILA(Text Line)	**98.47**	95.88	96.21	97.46	95.26	96.68	**99.16**	89.67
H-VILA(Text Block)	98.01	96.45	96.14	97.38	96.31	96.33	99.08	91.67
# Tokens in Class	395788	88531	90775	26742	7083	223739	7567934	22289

contd.	Copyright	Correspondence	Dates	Editor	Equation	Figure	Glossary	Keywords
$BERT_{BASE}$	97.34	89.66	94.56	99.71	17.60	94.05	80.18	93.42
$BERT_{BASE}$ + **I-VILA**(Text Line)	97.38	89.57	94.60	99.93	25.00	94.84	81.35	94.34
$BERT_{BASE}$ + **I-VILA**(Text Block)	97.85	91.29	94.99	99.95	29.46	95.52	80.45	95.40
$LayoutLM_{BASE}$	97.63	89.99	94.80	99.90	30.78	95.52	83.83	94.95
$LayoutLM_{BASE}$ + Sentence Breaks	97.62	90.07	94.73	99.95	20.73	95.83	84.99	93.88
$LayoutLM_{BASE}$ + **I-VILA**(Text Line)	97.47	90.97	95.20	99.93	26.42	95.67	84.16	94.82
$LayoutLM_{BASE}$ + **I-VILA**(Text Block)	97.66	91.04	95.13	**100.00**	**39.28**	95.74	**87.00**	**96.23**
Simple Group Classifier	97.56	**92.11**	**95.47**	**100.00**	33.17	95.77	80.35	95.64
H-VILA(Text Line)	97.78	89.96	94.98	99.91	15.60	95.63	84.01	93.69
H-VILA(Text Block)	**97.98**	90.37	94.92	100.00	30.64	**95.86**	78.29	96.15
# Tokens in Class	57419	26653	23702	2937	761	581554	2807	7012

contd.	Page Number	References	Table	Title	Type	Unknown	Macro F1
$BERT_{BASE}$	98.32	99.60	94.11	97.60	87.62	88.60	90.78
$BERT_{BASE}$ + **I-VILA**(Text Line)	98.82	99.60	94.53	97.77	93.70	88.14	91.65
$BERT_{BASE}$ + **I-VILA**(Text Block)	98.92	99.64	94.31	98.19	93.09	88.81	92.31
$LayoutLM_{BASE}$	98.94	99.62	95.30	97.91	91.24	89.19	92.34
$LayoutLM_{BASE}$ + Sentence Breaks	98.90	99.61	95.63	98.13	91.68	89.14	91.83
$LayoutLM_{BASE}$ + **I-VILA**(Text Line)	**99.05**	99.63	95.61	97.80	94.59	89.86	92.37
$LayoutLM_{BASE}$ + **I-VILA**(Text Block)	**99.05**	99.65	95.73	**98.39**	**95.17**	**90.47**	**93.38**
Simple Group Classifier	99.02	99.61	93.94	98.18	94.91	89.60	92.65
H-VILA(Text Line)	98.96	99.63	**96.02**	97.76	93.61	90.00	91.65
H-VILA(Text Block)	99.16	**99.68**	95.00	98.36	95.07	89.23	92.37
# Tokens in Class	46884	2340796	558103	22110	4543	54639	—

Table 7: Prediction F1 breakdown for all models on the GROTOAP2 dataset.

	Abstract	Author	Caption	Date	Figure	Footer	List	Paragraph	Reference	Section	Table	Title	Macro F1
$BERT_{BASE}$	97.82	89.96	93.91	87.33	71.97	84.76	75.99	96.84	92.05	92.81	74.19	89.31	87.24
$BERT_{BASE}$ + **I-VILA**(Text Line)	97.99	90.67	95.74	88.12	88.85	88.29	80.20	97.85	92.68	94.91	77.39	90.34	90.25
$BERT_{BASE}$ + **I-VILA**(Text Block)	98.15	90.66	96.56	87.83	79.49	88.40	80.72	97.51	92.62	94.86	76.91	90.22	89.49
$LayoutLM_{BASE}$	98.63	92.25	96.88	87.13	76.56	94.26	89.67	97.72	93.16	96.31	77.38	92.80	91.06
$LayoutLM_{BASE}$ + Sentence Breaks	98.48	92.70	96.93	88.06	77.65	94.35	90.46	97.81	92.61	96.58	78.84	92.81	91.44
$LayoutLM_{BASE}$ + **I-VILA**(Text Line)	98.57	92.64	97.35	87.87	**90.78**	94.37	90.77	98.44	92.87	**96.60**	80.43	92.78	92.79
$LayoutLM_{BASE}$ + **I-VILA**(Text Block)	**98.68**	92.31	97.44	87.69	83.41	94.03	90.56	98.13	93.27	96.44	79.51	92.48	92.00
$LayoutLMv2_{BASE}$	**98.68**	**93.04**	**97.49**	**89.55**	85.60	**95.30**	**93.63**	**98.46**	**94.30**	96.48	**84.41**	93.10	**93.34**
Simple Group Classifier	93.85	84.68	96.55	71.04	80.63	91.58	83.84	97.53	92.54	85.33	73.85	92.65	87.01
H-VILA(Text Line)	**98.68**	90.95	95.46	80.99	88.79	93.84	90.77	98.36	93.81	95.27	78.46	89.81	91.27
H-VILA(Text Block)	98.57	86.81	95.76	70.33	80.29	91.23	79.82	97.53	92.97	86.70	79.84	**93.52**	87.78
# Tokens in Class	461898	81061	858862	3275	932150	158176	684786	20630188	1813594	154062	235801	26355	—

Table 8: Prediction F1 breakdown for all models on the DocBank dataset.