Hierarchical Multimodal Pre-training for Visually Rich Webpage Understanding

Hongshen Xu
xuhongshen@sjtu.edu.cn
X-LANCE lab
Shanghai Jiao Tong University
Shanghai, China

Lu Chen*
chenlusz@sjtu.edu.cn
X-LANCE lab
Shanghai Jiao Tong University
Shanghai, China

Zihan Zhao
zhao_mengxin@sjtu.edu.cn
X-LANCE lab
Shanghai Jiao Tong University
Shanghai, China

Da Ma
mada123@sjtu.edu.cn
MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
Shanghai, China

Ruisheng Cao
211314@sjtu.edu.cn
MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
Shanghai, China

Zichen Zhu
JamesZhutheThird@sjtu.edu.cn
MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
Shanghai, China

Kai Yu*
kai.yu@sjtu.edu.cn
MoE Key Lab of Artificial Intelligence
Shanghai Jiao Tong University
Shanghai, China

ABSTRACT

The growing prevalence of visually rich documents, such as webpages and scanned/digital-born documents (images, PDFs, etc.), has led to increased interest in automatic document understanding and information extraction across academia and industry. Although various document modalities, including image, text, layout, and structure, facilitate human information retrieval, the interconnected nature of these modalities presents challenges for neural networks. In this paper, we introduce WebLM, a multimodal pre-training network designed to address the limitations of solely modeling text and structure modalities of HTML in webpages. Instead of processing document images as unified natural images, WebLM integrates the hierarchical structure of document images to enhance the understanding of markup-language-based documents. Additionally, we propose several pre-training tasks to model the interaction among text, structure, and image modalities effectively. Empirical results demonstrate that the pre-trained WebLM significantly surpasses previous state-of-the-art pre-trained models across several webpage understanding tasks. The pre-trained models and code are available at https://github.com/X-LANCE/weblm.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; • **Information systems** → **Web mining**.

KEYWORDS

multimodal pre-training, visually rich document understanding, web reading comprehension

ACM Reference Format:
Hongshen Xu, Lu Chen, Zihan Zhao, Da Ma, Ruisheng Cao, Zichen Zhu, and Kai Yu. 2024. Hierarchical Multimodal Pre-training for Visually Rich Webpage Understanding. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3616855.3635753

1 INTRODUCTION

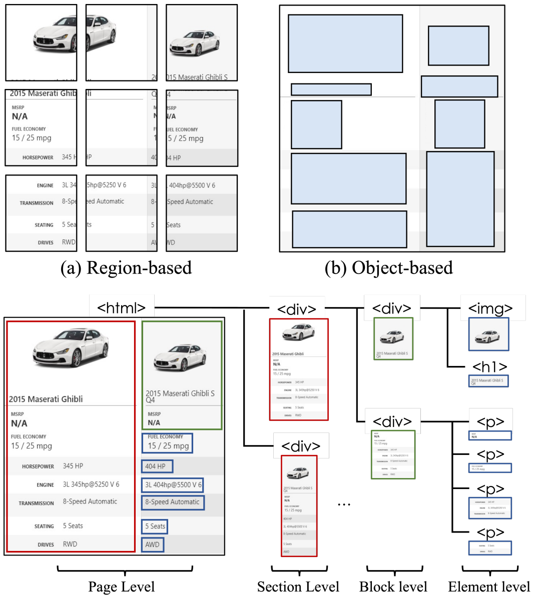
Visually rich documents have become the primary means of organizing, presenting, storing, and transmitting information over the Internet for billions of individuals. Recent advancements in the domain of deep learning and natural language processing have led to increasing attention toward automatically understanding and extracting information from these documents due to their diverse application scenarios [10, 18, 19, 30, 31]. To address the challenges posed by the cross-modality interconnections within visually rich documents, self-supervised training on large-scale unlabeled data [6, 8, 26] and multimodal pre-training techniques [5, 29, 33, 34] have emerged as promising approaches for Visually Rich Document Understanding (VRDU) tasks.

Multimodal pre-training models for documents can be broadly classified into two categories, namely image-oriented and text-oriented, based on the target document type. Image-oriented methods [2, 11, 17, 23, 37, 38] deal with scanned/digital-born documents, where document images are easily accessible, and textual portions

*Lu Chen and Kai Yu are the corresponding authors.

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635753



(a) Region-based

(b) Object-based

(c) Our **structure-based** hierarchical visual feature

Figure 1: Comparison among different multimodal document pre-training methods.

are typically acquired using external optical character recognition (OCR) tools. Text-oriented methods [7, 12, 13, 21], on the other hand, are concerned with markup-language-based documents such as webpages. Existing webpage pre-training models only use HTML as input, which consists of the structured description language and natural language content, i.e., the structure and text modalities. This situation arises mainly because the current web pre-training datasets are either HTML-only like Common Crawl [1], or failed to reach the pre-training scale. However, it is impossible to understand real-world webpages using HTML alone due to the lack of information from other resources as we discussed in §3.1.

Furthermore, it is hard to seamlessly apply current multimodal pre-training methods to web pages. These approaches often treat document images as natural images, neglecting the structural complexities inherent in documents. As depicted in Figure 1(a), **region-based** methods like LayoutLMv3[17] partitions images into regions to extract region-level features, while **object-based** methods[23] in Figure 1(b) rely on external tools to identify document objects and then extract object-level features. Unfortunately, both two types of methods neither capture multi-granularity visual features nor model the semantic relationship among those features. On the one hand, webpages exhibit a hierarchical structure, ranging from pages to sections, regions, and elements. External tools such as optical character recognition (OCR) or object detectors only recognize objects at a specific granularity, yet they are unable to capture features across various levels. On the other hand, there exist diverse semantic relationships between visual features of webpages, such as the sibling relationship between two elements or the parent-child relationship between the element and its parent section. The key to modeling such relationships is the structure of webpages that previous methods struggle to encode.

To address the above problems, we first collect a large-scale multimodal dataset for webpage pre-training, comprising a collection of **6 million** webpages from over **60,000** domains. This dataset encompasses HTML code, screenshots, and corresponding metadata. Second, we propose **WebLM**, a unified Transformer framework that concurrently models text, structure (markup language), and image modalities for understanding webpages. As shown in Figure 1(c), WebLM is able to extract hierarchical visual features with the incorporation of HTML structure. This is implemented by considering the visual alignment between HTML tags and image regions contained in the metadata of our datasets. Last, we propose two novel pre-training tasks: **Tree Structure Prediction (TSP)**, focuses on predicting the tree-relationship between HTML nodes, which models the semantic structure of webpages both textually and visually; **Visual Misalignment Detection(VMD)**, incorporates noise into the image region of HTML tags, compelling the model to be robust to the visual alignment between the two modalities.

We evaluate the WebLM models on the Web-based Structural Reading Comprehension (WebSRC) [4] dataset and the Structured Web Data Extraction (SWDE) [14]dataset. Experimental results show that our WebLM significantly outperforms previous SOTA pre-trained models. Ablation studies further demonstrate the effectiveness of incorporating the hierarchical visual feature.

The contributions of this paper are summarized as follows:

- We collect a large-scale multimodal dataset of **6 million** webpages from over **60,000** domains. The dataset, pre-trained models, and code are publicly available.
- We propose WebLM for webpage understanding, which introduces hierarchical visual features by first incorporating HTML structure into visual feature extraction.
- We propose two novel pre-training tasks to effectively model the semantic structure of webpages and enhance the visual robustness of WebLM.

2 RELATED WORK

2.1 Multi-modal Document Pre-training

Visually rich documents can be roughly divided into two categories based on the modalities involved: one is **image-centric** documents with image modality at the core, such as receipts and PDFs, where tasks often provide only image information and require external tools like OCR to obtain text and its location; the other is **text-centric** web documents, where the document image needs to be interactively and dynamically rendered based on the markup-language-based documents such as HTML/XML.

For scanned/digital-born documents, current pre-training methods often focus on extracting different granularity of visual features and then modeling the modality interaction via pre-training tasks. Both LayoutLM [37] and LayoutLMv2 [38] uses ResNet-101 to extract fine-grained token-level visual features, whereas LayoutLMv2 introduces Text-Image Alignment and Text-Image Matching tasks to enhance the modality interaction. LayoutLMv3 [17] and DiT [22] adopt the encoding approach of Image Transformers (such as ViT [9]) for document image encoding. However, previous methods always overlook the hierarchical structure of documents, and the association between images and text is often provided by off-the-shelf tools, making it difficult for the model to learn.

[1] https://commoncrawl.org/

For mark-language-based documents represented by webpages, existing pre-training models mainly focus on encoding the HTML source code, emphasizing the interaction between textual and structure modalities. MarkupLM [21] inputs the text token sequence of HTML code and incorporates the xpath of each text's node as relation embedding. DOM-LM [7] extracts various structured information for text tokens, such as depth, tag type, and node index. Webformer [12] designs a recursive encoding method for the tree structure of webpages. Furthermore, HTLM [1] focuses on zero-shot prompting through HTML-based pre-training. Those pre-training models mainly utilize HTML as inputs while ignoring the image modality. However, as discussed in §3.1, HTML code in webpages contains only a portion of the information, while more style and structural information are found in webpage screenshots.

2.2 Webpages Understanding

Webpages are the primary means for people to store, display, and transmit information on the Internet, making the automatic understanding and information extraction of webpages a blooming research topic. Hao et al. [14] proposed the SWDE dataset for information extraction on webpages. Tanaka et al., as well as Chen et al., introduced web-oriented reading comprehension datasets VisualMRC [35] and WebSRC [4], respectively, requiring models to understand the spatial structure of webpages as well as the textual content to answer corresponding questions. At the same time, many approaches [27, 32, 39] employ graph neural networks to encode node relationships in webpages. Additionally, large language models [13] have been proven to possess strong webpage understanding capabilities via few-shot learning.

3 WEBLM

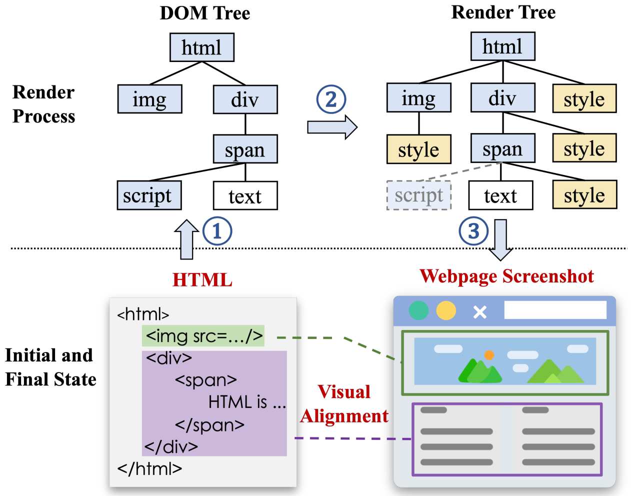
WebLM is a multimodal pre-training framework that incorporates HTML structure into visual feature extraction. The motivation of WebLM is discussed in §3.1, followed by the introduction of model architecture (§3.2) and pre-training tasks (§3.3).

3.1 Motivation of WebLM

We show the overall rendering process of webpages in Figure 2. As we can see, it is impossible to represent and understand real-world webpages using HTML alone. This is mainly due to two reasons:

- HTML does not contain information from external files, such as JavaScript, CSS, images, and other resources.
- Even with all the resources, rendering a webpage still requires browsers to interpret and execute the code, a complex process that existing models struggle to learn.

Therefore, we believe that understanding webpages necessitates a multimodal approach, simultaneously incorporating both HTML code and webpage screenshots. On the one hand, screenshots contain the most complete style information, while HTML encompasses all content information, ensuring the coverage of all essential information. On the other hand, HTML represents the initial state of the webpage, and the screenshot corresponds to the final state. Employing these two states for webpage understanding eliminates the need to learn browser rendering logic. Moreover, the completion of the rendering process results in a direct **visual alignment** between each node in the HTML and the region in the screenshot. WebLM leverages this alignment to provide visual features for each level of nodes in the HTML tree, effectively fusing structural and visual information to obtain hierarchical features of the images.



```
<html>
    <img src=.../>
    <div>
        <span>
            HTML is ...
        </span>
    </div>
</html>
```

Figure 2: The rendering process from HTML to a webpage.

3.2 Model Architecture

WebLM applies a unified multimodal Transformer to learn cross-modal representations where Figure 3 gives the architecture overview. The Transformer encoding layer is similar to BERT's [8], with key alterations made at the input layer. As shown in Figure 3, the input primarily comprises information from three modalities, corresponding to three different colors. Structure modality and content modality mainly come from HTML code, while visual modality comes from webpage screenshots. The primary design principles of the WebLM input layer are as follows:

- **Separate structure tokens from content tokens.** Previous models either directly take HTML code as input [4, 13], or solely input the textual token of HTML while considering structure modality as supplementary features [21]. In contrast, WebLM takes both structure and textual tokens of HTML as input while separating the two modalities. This approach not only retains full document structure information by preserving the structure tokens but also accelerates information flow within each modality.
- **Align visual features with HTML inputs.** Most multimodal pre-training methods [17, 23, 38]regard the extracted sequence of visual features as a separate input sequence with respect to text modalities. However, the alignment between text and image modalities is directly available in the context of webpages. Thus it is intuitive and reasonable to align visual features with text modalities before inputting them into the model rather than interacting during pre-training.

We further introduce the embedding details of each modality:

Token Embedding. To construct the input token sequence for the model, we first convert the HTML code into a DOM tree and traverse it in a depth-first order. During traversing, we perform **structural separate**, which places structure tokens and text tokens

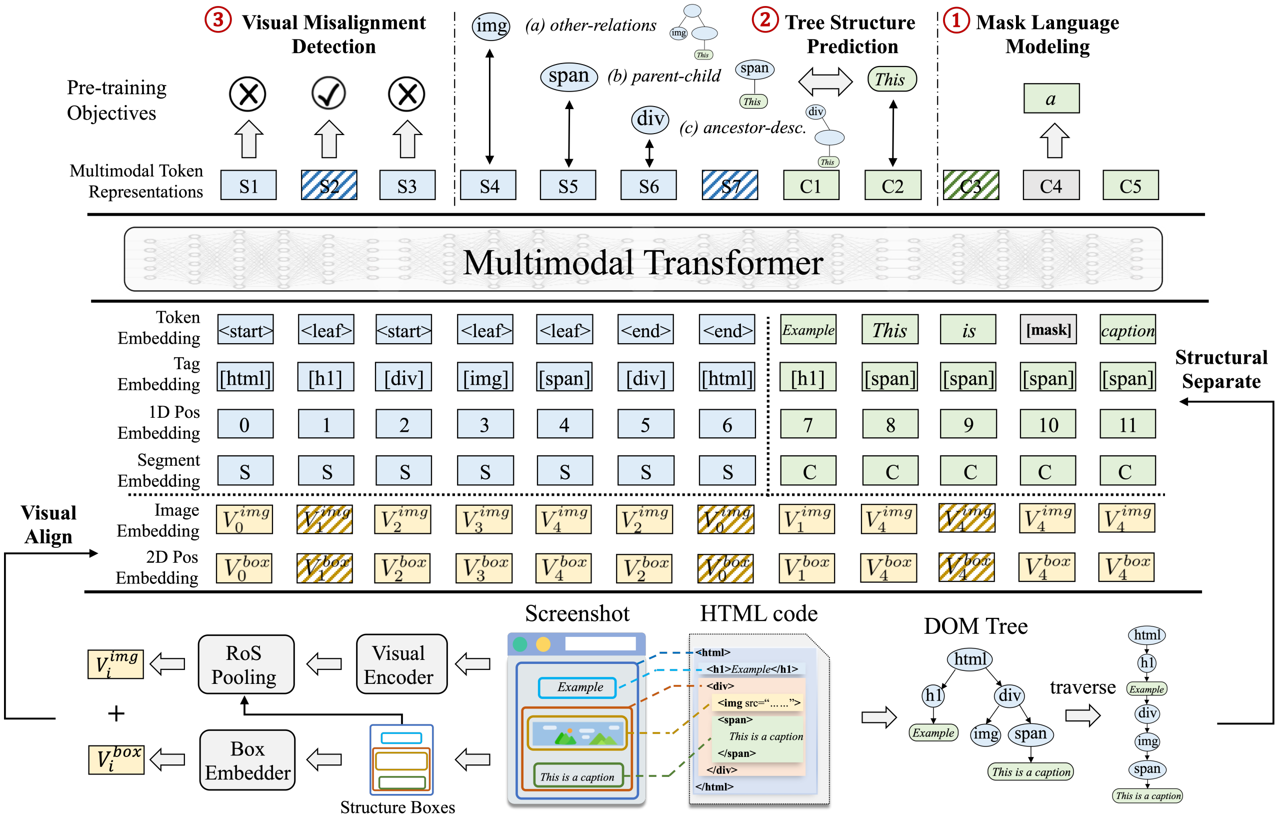


Figure 3: The architecture and pre-training objectives of WebLM. The blue and green parts represent structure and content inputs from HTML code respectively, while the yellow part corresponds to visual inputs from webpage screenshots.

in two lists. After concatenating the two sequences, we get the final input token sequence like

$$T = \{[\mathtt{CLS}], s_1, s_2, ..., [\mathtt{SEP}]c_1, c_2, ..., [\mathtt{SEP}]\}.$$

It is worth noting that our structure tokens only include the HTML tags themselves, excluding their attributes, as attribute information introduces significant noise. Furthermore, we simplify all tags into three types: <start_tag>, <end_tag>, and <leaf_tag>, and add tag types as additional embeddings. In fact, HTML tags have two main functions: one is to express the tree structure of HTML through the correspondence between pairs of opening and closing tags (e.g., <p> and </p>) and their nesting; the other is to indicate tag-specific roles, such as <h1> for headings and <img> for images. With approximately 120 common tags and extremely imbalanced distribution (e.g., the frequency of <div> tag is much higher than that of other tags.), separating structural and functional embeddings of tags allows for shared learning for the structural embedding of each tag, thus better modeling the structure of HTML.

Tag Embedding. There are approximately 120 common HTML tags, which can be classified according to their functions. Understanding the function of each tag helps to better interpret web content. Furthermore, each common tag has a textual description corresponding to its function, which can effectively aid the model in understanding the tag's purpose. Thus We employ sentence-Transformers [2] to extract an embedding vector from each tag's textual description [3] as the tag's initial embedding. For infrequent tags, we convert them to <unk> and initialize it randomly.

The final **text embedding** from HTML is the sum of four embeddings. Token embedding and 1D positional embedding represents the token and its index. Tag embedding represents the function of the token's corresponding HTML tag. For each token w_i from either structure input or content input, we incorporate the tag embedding based on the tag type of their respective nodes $tag_i \in \{\text{<html>}, \text{body}, \text{<div>}, ...\}$in the DOM tree. Besides, we use segment embedding to distinguish structure and text content tokens by assigning each token to a segment $seg_i \in \{[\mathtt{S}], [\mathtt{C}]\}$, Formally, we have the i-th text embedding for a token w_i:

$$\mathbf{t}_i = \text{TokEmb}(w_i) + \text{TagEmb}(tag_i) + \text{PosEmb1D}(i) + \text{SegEmb}(seg_i).$$

Image Embedding. WebLM employs a ResNeXt-FPN [25, 36] architecture as the backbone of the visual encoder. Given a webpage screenshot I, it is resized to 224 × 224 and fed into the visual backbone. Then WebLM extracts the corresponding visual feature for all the nodes on the HTML DOM tree. Specifically, while object detection models perform pooling on RoI (Regions of Interest) [16], WebLM conducts pooling on **RoS** (Regions of Structure nodes on the DOM tree) according to the visual alignment information of

[2] https://www.sbert.net/

[3] https://www.w3schools.com/tags/default.asp

each node. After obtaining the image embeddings for each node on the DOM tree, we perform **visual align** by aligning the node visual feature sequence with the input token sequence based on the correspondence between token and node, allowing each token to obtain its associated visual features. The image embedding of i-th token with the corresponding node n_i is formulated as

$$\mathrm{v}_i^{img} = \mathrm{RoSPool}(\mathrm{VisualEncoder}(\mathrm{I}))_{n_i}$$

2D Position Embedding. The image feature extracted by the visual encoder primarily contains style information, such as color, and font. In addition, we input the bounding box of the node region to embed spatial layout information. Following previous works [38], we normalize and discretize all coordinates to integers in the range $[0, 1000]$, and use two embedding layers to embed x-axis features and y-axis features separately. Given the i-th token and the normalized bounding box of its corresponding node n_i $\mathrm{box}_{n_i} = (x_0, x_1, y_0, y_1, w, h)$, we calculate the 2D position embedding by concatenating six bounding box features and aligning it to i-th token:

$$\mathrm{v}_i^{box} = \mathrm{Concat}(\mathrm{PosEmb2D_x}(x_0, x_1, w), \mathrm{PosEmb2D_y}(y_0, y_1, h))$$

The final **visual embedding** of the i-th token is the sum of its image embedding and 2D position embedding

$$\mathrm{v}_i = \mathrm{v}_i^{img} + \mathrm{v}_i^{box},$$

and we obtain the final input embedding x_i of the i-th token by adding its text embedding t_i and visual embedding v_i.

3.3 Pre-training Objectives

To efficiently model the complex structure of webpages and enhance the information exchange among the three modalities, We design three self-supervised pre-training tasks for WebLM, including mixed-modality MLM and cross-modality TSP and VMD tasks.

Masked Language Modeling (MLM). Following previous works [8, 37], we use MLM to enhance the model's language understanding capabilities. We randomly replace some content tokens with `[MASK]` and require the model to predict the original words. Unlike the Masked Visual-Language Modeling in LayoutLMv2 [38], we do not mask the corresponding regions of tokens in the image due to the unavailability of token-level region positions.

Tree Structure Prediction (TSP). We propose the TSP task based on the following two observations:

- While the structural separation facilitates intra-modality information flow, it slows down cross-modality information flow between structure and content modalities.
- The tree structure of HTML explicitly conveys the main semantic structure of both textual and visual inputs.

Thus the TSP task requires WebLM to predict the tree relationship between structure and content inputs to accelerate cross-modality information flow as well as modeling the semantic structure of webpages. Specifically, we sample a node token from the structural input and a text token from the content input, requiring the model to determine their relationship based on the DOM Tree $R \in \{$ `parent-child`, `ancestor-descendent`, `other-relations`$\}$. In addition, text tokens within the tree solely constitute leaf nodes, yet their quantity far surpasses that of structural nodes. Structural separation helps TSP to sample more diverse tree node pairs while simplifying the implementation via complex sampling algorithms.

Visual Misalignment Detection (VMD). As we described in sec 3.1, visual alignment between HTML nodes and screenshot regions is the key to constructing the multimodal input sequence of WebLM. Due to rendering issues or external interference, such alignment might introduce noise, potentially impacting the model's performance. Therefore, the TIM task is proposed to enhance the visual robustness. Specifically, we randomly sample some positions from the input sequence (including both structural and content inputs) and add noise to their corresponding visual feature regions, either enlarging or reducing them by 50%. The model is then asked to identify if the image information at each token has been affected by noise. This perturbing simultaneously alters both the 2D position embedding and image embedding, requiring the model to make judgments based on the textual modality or the semantic relationships between with surrounding tokens.

4 EXPERIMENTS

WebLM focuses on webpage understanding through multimodal pre-training. Therefore, we evaluate it on web-based question answering and information extraction tasks and further investigate the importance of each component through ablation studies.

4.1 Data

4.1.1 Pre-training Data.

Common Crawl. Common Crawl is a publicly available web crawl dataset that collects webpages from the internet. Instead of using the source code, we collect webpage links from Common Crawl. We traverse all links in a dataset snapshot [4] and categorize and sort them based on the domain name. We select 60,000 domains with the largest number of webpages, with each domain containing 100 webpages. We also use `fasttext` [20] to filter out non-English pages with an English classification score < 0.6. We then employ `Selenium` [5] to crawl the corresponding HTML, webpage screenshots, and bounding box information for each HTML node. Finally, we obtain a dataset of 6 million webpages for WebLM pre-training.

Pre-processing. Due to a large number of textual tokens of real-world webpages and the typically long screenshots, it is challenging to input entire webpages into the model for pre-training. Consequently, we construct pre-training features mainly through two approaches: simplifying the HTML input and extracting input segments from complete webpages. We only retain nodes on the HTML rendering tree that either contain text or have corresponding regions in the image, while removing nodes that do not have either (e.g., `<script>`, `<style>`, etc.). Furthermore, we simplify the HTML structure by replacing a node with its child node if it has only one child. After these modifications, we traverse the HTML tree and search for nodes whose total number of tokens in their child nodes and text lies within a certain range (i.e., between 128 and 512.) as input features. By extracting the corresponding screenshot area and HTML code segment, we construct features that meet input

[4] https://commoncrawl.org/2022/08/august-2022-crawl-archive-now-available/
[5] https://www.selenium.dev/

Table 1: Evaluation results on WebSRC. EM, F1, POS denotes the exact match score, the token level F1 score, and the path overlap score, respectively. We submit the models to the official of WebSRC for testing. * denotes reproduction results.

	Method	Modalities	Dev			Test		
			EM↑	F1↑	POS↑	EM↑	F1↑	POS↑
BASE	T-PLM(BERT) [4]	Text	52.12	61.57	79.74	39.28	49.49	67.68
	H-PLM(BERT) [4]	Text + HTML	61.51	67.04	82.97	52.61	59.88	76.13
	V-PLM(BERT) [4]	Text + HTML + Image	62.07	66.66	83.64	52.84	60.80	76.39
	DOM-LM [7]	Text + HTML	69.70	73.90	-	-	-	-
	LayoutLMv3* [17]	Text + Image	66.33	71.46	85.27	48.33	51.64	71.02
	MarkupLM [21]	Text + HTML	68.39	74.47	87.93	-	-	-
	MarkupLM*	Text + HTML	68.99	74.55	88.40	60.43	67.05	80.55
	WebLM	Text + HTML + Image	**72.14**	**79.67**	**89.36**	**65.95**	**72.30**	**83.77**
LARGE	T-PLM(Electra) [4]	Text	61.67	69.85	84.15	56.32	72.35	79.18
	H-PLM(Electra) [4]	Text + HTML	70.12	74.14	86.33	66.29	72.71	83.17
	V-PLM(Electra) [4]	Text + HTML + Image	73.22	76.16	87.06	68.07	75.25	84.96
	LayoutLMv3* [17]	Text + Image	71.38	75.73	87.74	57.68	63.33	79.76
	MarkupLM [21]	Text + HTML	74.43	80.54	90.15	-	-	-
	MarkupLM*	Text + HTML	73.38	79.83	89.93	69.09	76.45	87.24
	WebLM	Text + HTML + Image	**78.40**	**84.24**	**91.54**	**72.01**	**78.66**	**88.33**

length constraints. Additionally, if combined sibling nodes satisfy the input length limits, we also use their corresponding HTML code segments and largest bounding box screenshots as input features.

4.1.2 Fine-tuning Data.

WebSRC. WebSRC [4] is a Web-based Structural Reading Comprehension dataset that aims to test the ability of models to understand the contents of webpages as well as their structures. WebSRC consists of 400K question-answer pairs, which are collected from 6.4K webpages. Each question in WebSRC requires a certain structural understanding of a webpage to answer, and the answer is either a text span on the webpage or yes/no. Following the original paper, we use **Exact match (EM)**, **F1 score (F1)**, and **Path overlap score (POS)** as evaluation metrics.

SWDE. The Structured Web Data Extraction (SWDE) [14] dataset is a real-world collection of webpages used for automatic information extraction. It consists of 8 verticals, 80 websites (10 per vertical), and 124,291 webpages in total. The goal is to extract values corresponding to given attributes from a webpage, such as the *price* value in *shopping* pages. We use **page-level F1** scores as our evaluation metric as in previous works [21, 24, 41]. We follow MarkupLM to train and evaluate each vertical independently. In each vertical, we select k consecutive seed websites for training and use the remaining $10 - k$ websites for testing. The final results are obtained by averaging across all 8 verticals and all 10 permutations of seed websites per vertical, resulting in 80 experiments for each k.

4.2 Experiment Setup

Pre-training. The token-masked probability in MLM and visually noise-adding probability in VMD are both 15%. The probability of the bounding box increasing or decreasing in size is each 50%. The max number of selected node pairs is 1,000 in TSP for each sample, and we limit the ratio of pairs with `other-relations` as 60% to make a balance. We initialize WebLM from RoBERTa and train the base and large model for 300K steps on 8 NVIDIA A10 and A100 GPUs, respectively. For the ResNeXt-FPN part in the visual embedding layer, the backbone of a Mask-RCNN [15] model trained on PubLayNet [40] is leveraged [6]. We set the total batch size as 256, the learning rate as 5e-5, the max sequence length as 512, and the warmup ratio as 0.1. The selected optimizer is AdamW [28], with $\epsilon = 1e - 6$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, `weight decay` = 0.01, and a linear decay learning rate scheduler with 6% warmup steps. We also apply `FP16` to reduce GPU memory consumption and accelerate training.

Fine-tuning. We treat the WebSRC task and SWDE task as an extractive QA task and token classification task, respectively. In the input layer, we truncate any structural tokens exceeding a fixed length and concatenate the HTML text and the question as the content input. When the content length surpasses the limit, a sliding window mechanism is employed for multiple inputs. For separators such as [CLS] and [SEP], as well as question tokens, we consider them to be directly connected to the <html> node. For WebSRC, we fine-tune WebLM for 2 epochs with a total batch size of 64 and a learning rate of 1e-5. For SWDE, we fine-tune WebLM with 10 epochs, a total batch size of 64, and a learning rate of 2e-5. The warmup ratio is set to 0.1 and the max sequence length is set as 512 in both tasks, and we keep other hyper-parameters as default.

4.3 Baselines

We only introduce the SOTA pretrained models here and refer readers to [4, 41] for more details above non-pretrained baselines: **DOM-LM.** DOM-LM[7] is an HTML-based pre-trained model that takes text tokens and several HTML DOM tree features as inputs, such as depth, tag type, and node index.

[6] "MaskRCNN ResNeXt101 32x8d FPN 3X" setting in https://github.com/hpanwar08/detectron2

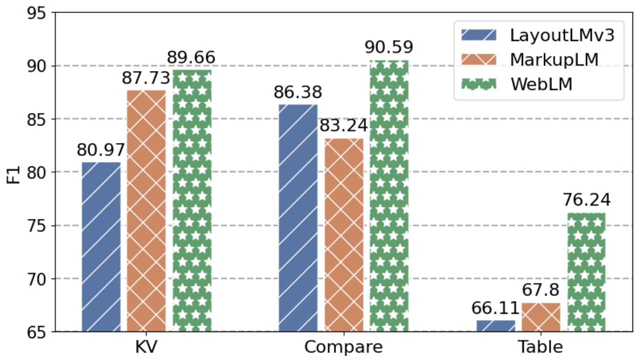


Figure 4: The performance comparison on different types of websites of WebSRC development set.

Table 2: Results on SWDE using different numbers of seed sites $k = \{1, 2, 3, 4, 5\}$. The baseline results are from [41].

Model \ #Seed Sites	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
SSM [3]	63.00	64.50	69.20	71.90	74.10
Render-Full [14]	84.30	86.00	86.80	88.40	88.60
FreeDOM-NL [24]	72.52	81.33	86.44	88.55	90.28
FreeDOM-Full [24]	82.32	86.36	90.49	91.29	92.56
SimpDOM [41]	83.06	88.96	91.63	92.84	93.75
MarkupLM$_{\text{BASE}}$	82.11	91.29	94.42	95.31	95.89
WebLM$_{\text{BASE}}$	**84.21**	**93.17**	**95.68**	**96.17**	**96.78**
MarkupLM$_{\text{LARGE}}$	85.71	93.57	96.12	96.71	97.37
WebLM$_{\text{LARGE}}$	**87.57**	**94.89**	**97.25**	**97.54**	**98.10**

LayoutLMv3. LayoutLMv3[17] is a multimodal pre-trained model for document understanding. It simplifies LayoutLMv2[38] by using patch embeddings (as in ViT) instead of leveraging a CNN backbone. LayoutLMv3 exhibits a general capacity for visual understanding while having a modest performance on textual modeling.
MarkupLM. MarkupLM[21] is a SOTA webpage pre-trained model which only inputs the text token sequence of HTML code and incorporates the xpath of each text's node as supplementary information. Instead of explicitly modeling the structure of HTML, it regards the tree relationship of node pairs as a type of relation embedding between text tokens. Consequently, MarkupLM achieves the best text-understanding abilities among all models.

4.4 Main Results

As shown in Table 1, both base and large versions of our proposed WebLM significantly outperform all baseline models on WebSRC dataset. Compared to MarkupLM, WebLM still exhibits substantial performance improvements. This demonstrates that WebLM can effectively utilize the information from all three modalities, achieving a better understanding of both webpage structure and textual content. Additionally, although LayoutLMv3 is not pre-trained on web data, it still exhibits good performance on the dev set, highlighting the importance of visual modality. However, its lower performance on the test set indicates a weaker generalization ability, emphasizing the necessity of pre-training on web data.

Table 3: Ablation study of pre-training tasks on WebSRC dev set.

Pre-training Data	Objectives			Metrics		
Samples	MLM	TSP	TIM	EM	F1	POS
1M	✓			64.17	72.13	86.33
1M	✓	✓		66.99	74.92	87.78
1M	✓	✓	✓	67.43	76.93	88.60
6M	✓	✓	✓	72.14	79.67	89.36

We further compare the model performance on different types of websites as shown in Figure 4. KV-type websites emphasize the comprehension of textual semantics, whereas Compare and Table-type websites underscore the significance of webpage sematic structure. We find that WebLM demonstrates a dual proficiency encompassing strong textual comprehension and better webpage structure modeling. Especially in visually complex webpages, i.e., Compare and Table-type websites, WebLM significantly outperforms the other two models. This success further demonstrates the necessity and effectiveness of introducing the hierarchical visual feature.

The results for the SWDE dataset are shown in Table 2. Since the SWDE dataset was created earlier, many webpages in the dataset do not contain visual information such as CSS and screenshots. Therefore, we render the HTML files in a browser to generate corresponding screenshots, which introduces a considerable amount of noise. Nevertheless, from the experimental results, we can observe that both the base and large models of our WebLM outperform MarkupLM. The performance improvement is more pronounced when the training data is limited, i.e., when k is small. This demonstrates that WebLM possesses robustness to visual information noise, maintaining a good webpage understanding capability even in noisy environments.

4.5 Ablation Study

4.5.1 Pre-training Tasks. The ablation results of pre-training tasks are shown in Table 3. We found that both TSP and VMD, two pre-training tasks focusing on inter-modal interactions, significantly contribute to the model's performance. When removing the TSP task, which focuses on the interaction between structure and content modalities, WebLM's performance loss is greater, with a decrease of 2.8 points in both the EM score and F1 score, demonstrating that modeling HTML structure can better help the model understand web content. The VMD task enhances model performance by strengthening the interaction between text and image modalities. Furthermore, our WebLM pre-trained with just 1 million webpages performs on par with the MarkupLM model trained with 24 million webpages, which demonstrates the efficient utilization of our high-quality datasets.

4.5.2 Visual Features. The ablation study results of different visual embeddings are shown in Table 4. We find that both Image embedding and 2D position embedding have a strong impact on the model's performance. When these two features are removed, the model's performance experiences a sharp decline. Moreover, Image embedding has a greater influence on predicting the EM score,

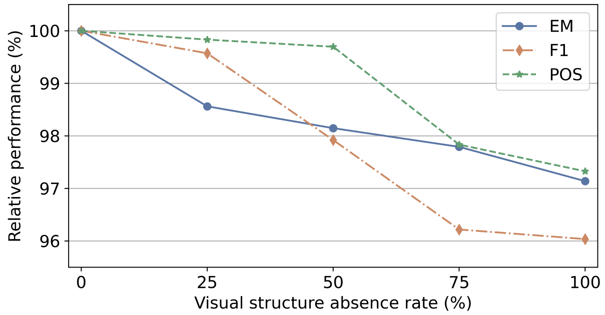


Figure 5: Relative performance on WebSRC Dev set compared to full-structured methods after truncating HTML structures.

Table 4: Ablation study of visual embeddings on the WebSRC dev set.

Method	EM↑	F1↑	POS↑
WebLM$_{BASE}$+MLM&TSP	66.99	74.92	87.78
-w/o Image Embedding	60.56$_{(-6.43)}$	71.77$_{(-3.15)}$	85.81$_{(-1.97)}$
-w/o 2D Position Embedding	62.91$_{(-4.08)}$	70.18$_{\mathbf{(-4.74)}}$	84.22$_{\mathbf{(-3.56)}}$
-w/o Visual Embedding	58.52$_{\mathbf{(-8.47)}}$	71.13$_{(-3.79)}$	84.54$_{(-3.24)}$

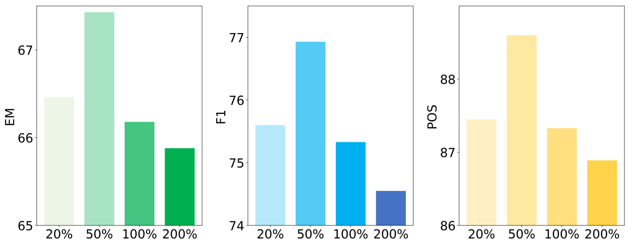


Figure 6: Experimental results on the WebSRC dev set while introducing different levels of visual noise in VMD.

while 2D position embedding has a more significant impact on predicting the F1 score. When both features are removed, the task's EM score experiences a more substantial decrease. This demonstrates the importance of visual features and further confirms WebLM's ability to effectively incorporate and utilize visual features.

4.5.3 Hierarchically Visual Structure. While previous work can also obtain features of each sub-region within an image, our method uniquely leverages the HTML structure to hierarchically combine these sub-region features. This generation of visual features is an attribute absent in prior studies. We also demonstrate its effectiveness with additional experiments. As shown in Figure 5, we truncate HTML non-leaf nodes that are below a specific depth (-25% signifies truncating those nodes that are less than 25% of the maximum depth). This approach allows for the preservation of all fine-grained visual information in the image while eliminating hierarchical visual features of other granularities. Our experiments demonstrate that even with the inclusion of visual information from images, there is a substantial decline in model performance if the incorporation of HTML structure is omitted.

4.5.4 Tree Structure Prediction. We also observe the contribution of different parts of the WebLM input to the prediction of HTML structure. By setting aside a portion of the test data, we tested the accuracy of the TSP task on this test set after training the model with different settings for 10,000 steps. "W/o Closing tag" refers to the situation where we have removed all closing tags from the HTML structure tokens. "w/o tag order" refers to the condition where, after removing these closing tags, we shuffle all structure nodes for input. These two ablation experiments are designed to observe how the model predicts the DOM Tree structure using the HTML structure input. Our results in Table 5 show that different features all contribute to predicting HTML structure and complement each other. Notably, visual feature plays a crucial role in modeling HTML structure as shown in the table.

Table 5: Ablation study of different input features on Tree Structure Prediction.

Method	TSP Accuracy
WebLM$_{BASE}$	99.44
-w/o closing tag	96.54
-w/o closing tag & tag order	95.38
-w/o visual feature	90.18
-w/o visual feature & closing tag & tag order	72.38

4.6 Impact of Various Noise Levels in VMD

Figure 6 shows the impact of the VMD pre-training task when different levels of noise are applied to the images. Our results show that the model performs best when the noise is either enlarging or reducing the image region by 50%. We believe that when the noise is too small, it does not help the model to learn the robustness of visual information and alignment of visual and textual information, whereas when the noise is too large, it interferes with the model's understanding and learning of visual features.

5 CONCLUSION

In this work, we address the automated webpage understanding and information extraction by incorporating hierarchical visual information through multimodal pre-training. We primarily leverage the structured correspondence between HTML code and corresponding webpage screenshots to construct input for WebLM and perform information fusion across different modalities by devising pre-training tasks. Extensive experiments demonstrate the effectiveness of the proposed architecture, and subsequent ablation studies further highlight the importance of visual information in the process of webpage understanding. In the future, we plan to apply the WebLM to scanned/digital-born documents. By conducting automated analysis and structure construction on these documents, we aim to address the hierarchical alignment problem between image and text modalities in such document scenarios.

ACKNOWLEDGMENTS

This work is funded by the China NSFC Projects (92370206, U23B2057, 62106142, and 62120106006) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

REFERENCES

[1] Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. Htlm: Hyper-text pre-training and prompting of language models. *arXiv preprint arXiv:2107.06955* (2021).

[2] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*. 993–1003.

[3] Andrew Carlson and Charles Schafer. 2008. Bootstrapping information extraction from semi-structured web pages. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part I 19*. Springer, 195–210.

[4] Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. Websrc: A dataset for web-based structural reading comprehension. *arXiv preprint arXiv:2101.09465* (2021).

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*. Springer, 104–120.

[6] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).

[7] Xiang Deng, Prashant Shiralkar, Colin Lockard, Binxuan Huang, and Huan Sun. 2022. DOM-LM: Learning Generalizable Representations for HTML Documents. *arXiv preprint arXiv:2201.10608* (2022).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[10] Filip Graliński, Tomasz Stanisławek, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2020. Kleister: A novel task for information extraction involving long documents with complex layout. *arXiv preprint arXiv:2003.02356* (2020).

[11] Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Nikolaos Barmpalios, Ani Nenkova, and Tong Sun. 2021. Unidoc: Unified pretraining framework for document understanding. *Advances in Neural Information Processing Systems* 34 (2021), 39–50.

[12] Yu Guo, Zhengyi Ma, Jiaxin Mao, Hongjin Qian, Xinyu Zhang, Hao Jiang, Zhao Cao, and Zhicheng Dou. 2022. Webformer: Pre-training with web pages for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1502–1512.

[13] Izzeddin Gur, Ofir Nachum, Yingjie Miao, Mustafa Safdari, Austin Huang, Aakanksha Chowdhery, Sharan Narang, Noah Fiedel, and Aleksandra Faust. 2022. Understanding HTML with Large Language Models. *arXiv preprint arXiv:2210.03945* (2022).

[14] Qiang Hao, Rui Cai, Yanwei Pang, and Lei Zhang. 2011. From one tree to a forest: a unified solution for structured web data extraction. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 775–784.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[17] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*. 4083–4091.

[18] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1516–1520.

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

[20] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[21] Junlong Li, Yiheng Xu, Lei Cui, and Furu Wei. 2021. Markuplm: Pre-training of text and markup language for visually-rich document understanding. *arXiv preprint arXiv:2110.08518* (2021).

[22] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. 2022. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3530–3539.

[23] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5652–5660.

[24] Bill Yuchen Lin, Ying Sheng, Nguyen Vo, and Sandeep Tata. 2020. Freedom: A transferable neural architecture for structured information extraction on web documents. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1092–1102.

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2117–2125.

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[27] Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. 2020. Zeroshotceres: Zero-shot relation extraction from semi-structured webpages. *arXiv preprint arXiv:2005.07105* (2020).

[28] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* 32 (2019).

[30] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2200–2209.

[31] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. CORD: a consolidated receipt dataset for post-OCR parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.

[32] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2018. Graphie: A graph-based framework for information extraction. *arXiv preprint arXiv:1810.13083* (2018).

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[34] Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490* (2019).

[35] Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13878–13888.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1492–1500.

[37] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. 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*. 1192–1200.

[38] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020. Layoutlmv2: Multimodal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740* (2020).

[39] Zihan Zhao, Lu Chen, Ruisheng Cao, Hongshen Xu, Xingyu Chen, and Kai Yu. 2022. TIE: Topological Information Enhanced Structural Reading Comprehension on Web Pages. *arXiv preprint arXiv:2205.06435* (2022).

[40] 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)*. IEEE, 1015–1022.

[41] Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. 2021. Simplified dom trees for transferable attribute extraction from the web. *arXiv preprint arXiv:2101.02415* (2021).