MarkupLM: Pre-training of Text and Markup Language for Visually Rich Document Understanding

Junlong Li[1]*, **Yiheng Xu**[2]*, **Lei Cui**[2], **Furu Wei**[2]
[1]Shanghai Jiao Tong University
[2]Microsoft Research Asia
lockonn@sjtu.edu.cn
{t-yihengxu,lecu,fuwei}@microsoft.com

Abstract

Multimodal pre-training with text, layout, and image has made significant progress for Visually Rich Document Understanding (VRDU), especially the fixed-layout documents such as scanned document images. While, there are still a large number of digital documents where the layout information is not fixed and needs to be interactively and dynamically rendered for visualization, making existing layout-based pre-training approaches not easy to apply. In this paper, we propose **MarkupLM** for document understanding tasks with markup languages as the backbone, such as HTML/XML-based documents, where text and markup information is jointly pre-trained. Experiment results show that the pre-trained MarkupLM significantly outperforms the existing strong baseline models on several document understanding tasks. The pre-trained model and code will be publicly available at https://aka.ms/markuplm.

1 Introduction

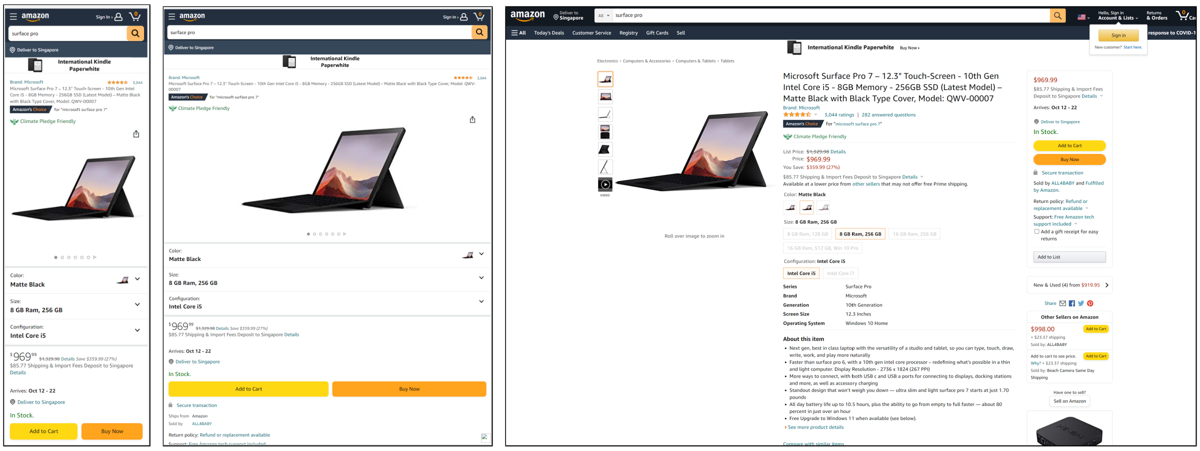
Multimodal pre-training with text, layout, and visual information has recently become the de facto approach (Xu et al., 2020, 2021a,b; Pramanik et al., 2020; Garncarek et al., 2021; Hong et al., 2021; Powalski et al., 2021; Wu et al., 2021; Li et al., 2021a,b; Appalaraju et al., 2021) in Visually-rich Document Understanding (VRDU) tasks. These multimodal models are usually pre-trained with the Transformer architecture (Vaswani et al., 2017) using large-scale unlabeled scanned document images (Lewis et al., 2006) or digital-born PDF files, followed by task-specific fine-tuning with relatively small-scale labeled training samples to achieve the state-of-the-art performance on a variety of document understanding tasks, including form understanding (Jaume et al., 2019; Xu et al., 2021b), receipt understanding (Huang et al., 2019; Park et al., 2019), complex document understanding (Graliński et al., 2020), document type classification (Harley et al., 2015), and document visual question answering (Mathew et al., 2021), etc. Significant progress has been witnessed not only in research tasks within academia, but also in different real-world business applications such as finance, insurance, and many others.

Visually rich documents can be generally divided into two categories. The first one is the fixed-layout documents such as scanned document images and digital-born PDF files, where the layout and style information is pre-rendered and independent of software, hardware, or operating system. This property makes existing layout-based pre-training approaches easily applicable to document understanding tasks. While, the second category is the markup-language-based documents such as HTML/XML, where the layout and style information needs to be interactively and dynamically rendered for visualization depending on the software, hardware, or operating system, which is shown in Figure 1. For markup-language-based documents, the 2D layout information does not exist in an explicit format but usually needs to be dynamically rendered for different devices, e.g., mobile/tablet/desktop, which makes current layout-based pre-trained models hard to apply. Therefore, it is indispensable to leverage the markup structure into document-level pre-training for downstream VRDU tasks.

To this end, we propose **MarkupLM** to jointly pre-train text and markup language in a single framework for markup-based VRDU tasks. Distinct from fixed-layout documents, markup-based documents provide another viewpoint for the document representation learning through markup structures because the 2D position information and document image information cannot be used straightforwardly during the pre-training. Instead, MarkupLM takes advantage of the tree-based markup

[*] Equal contributions during internship at Microsoft Research Asia. Corresponding authors: Lei Cui and Furu Wei

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6078 - 6087
May 22-27, 2022 ©2022 Association for Computational Linguistics



(a) Mobile (b) Tablet (c) Desktop

Figure 1: HTML-based webpages rendered by different platforms, such as mobile, tablet and desktop. (https://amzn.to/2ZZoi5R)

structures to model the relationship among different units within the document. Similar to other multimodal pre-trained layout-based models, MarkupLM has four input embedding layers: (1) a text embedding that represents the token sequence information; (2) an XPath embedding that represents the markup tag sequence information from the root node to the current node; (3) a 1D position embedding that represents the sequence order information; (4) a segment embedding for downstream tasks. The overall architecture of MarkupLM is shown in Figure 2. The XPath embedding layer can be considered as the replacement of 2D position embeddings compared with the LayoutLM model family (Xu et al., 2020, 2021a,b). To effectively pre-train the MarkupLM, we use three pre-training strategies. The first is the Masked Markup Language Modeling (MMLM), which is used to jointly learn the contextual information of text and markups. The second is the Node Relationship Prediction (NRP), where the relationships are defined according to the hierarchy from the markup trees. The third is the Title-Page Matching (TPM), where the content within "<title> ... </title>" is randomly replaced by a title from another page to make the model learn whether they are correlated. In this way, MarkupLM can better understand the contextual information through both the language and markup hierarchy perspectives. We evaluate the MarkupLM models on the Web-based Structural Reading Comprehension (WebSRC) dataset (Chen et al., 2021) and the Structured Web Data Extraction (SWDE) dataset (Hao et al., 2011). Experiment results show that the pre-trained MarkupLM significantly outperforms the several strong baseline models in these tasks.

The contributions of this paper are summarized as follows:

- We propose MarkupLM to address the document representation learning where the layout information is not fixed and needs to be dynamically rendered. For the first time, the text and markup information is pre-trained in a single framework for the VRDU tasks.
- MarkupLM integrates new input embedding layers and pre-training strategies, which have been confirmed effective on HTML-based downstream tasks.
- The pre-trained MarkupLM models and codes for fine-tuning will be publicly available at https://aka.ms/markuplm.

2 MarkupLM

MarkupLM utilizes the DOM tree in markup language and the XPath query language to obtain the markup streams along with natural texts in markup-language-based documents (Section 2.1). We propose this Transformer-based model with a new XPath embedding layer to accept the markup sequence inputs (Section 2.2) and pre-train it with three different-level objectives, including Masked Markup Language Modeling (MMLM), Node Relation Prediction (NRP), and Title-Page Matching (TPM) (Section 2.3).

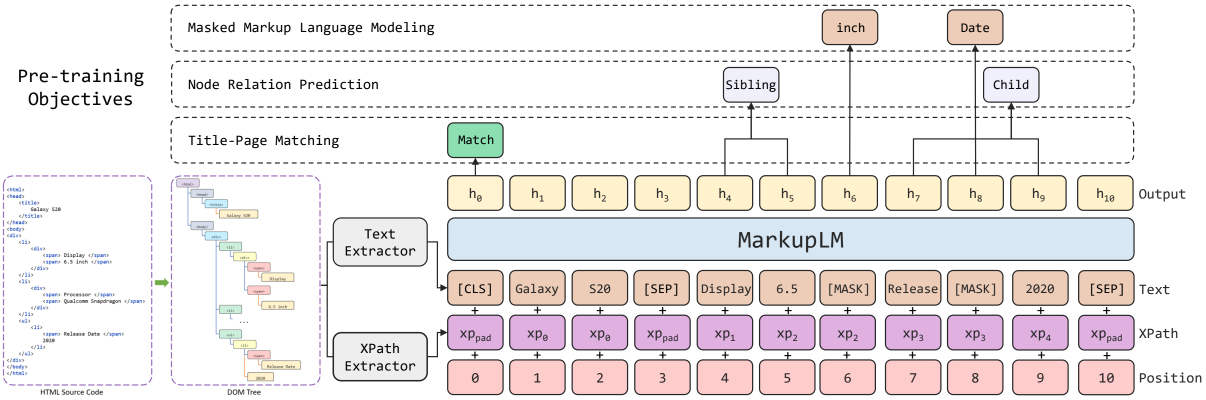


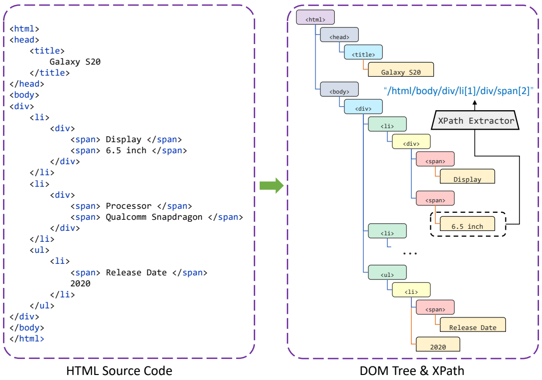
Figure 2: The architecture of MarkupLM, where the pre-training tasks are also included.

2.1 DOM Tree and XPath

A DOM[1] tree is the tree structure object of a markup-language-based document (*e.g.*, HTML or XML) in the view of DOM (Document Object Model) wherein each node is an object representing a part of the document.

XPath[2] (XML Path Language) is a query language for selecting nodes from a markup-language-based document, which is based on the DOM tree and can be used to easily locate a node in the document. In a typical XPath expression, like `/html/body/div/li[1]/div/span[2]`, the texts stand for the tag name of the nodes while the subscripts are the ordinals of a node when multiple nodes have the same tag name under a common parent node.

We show an example of DOM tree and XPath along with the corresponding source code in Figure 3, from which we can clearly identify the genealogy of all nodes within the document, as well as their XPath expressions.



```
<html>
<head>
    <title>
        Galaxy S20
    </title>
</head>
<body>
<div>
    <li>
        <div>
            <span> Display </span>
            <span> 6.5 inch </span>
        </div>
    </li>
    <li>
        <div>
            <span> Processor </span>
            <span> Qualcomm Snapdragon </span>
        </div>
    </li>
    <ul>
        <li>
            <span> Release Date </span>
            2020
        </li>
    </ul>
</div>
</body>
</html>
```

Figure 3: An example of DOM tree and XPath with the source HTML code.

2.2 Model Architecture

To take advantage of existing pre-trained models and adapt to markup-language-based tasks (*e.g.*, webpage tasks), we use the BERT (Devlin et al., 2019) architecture as the encoder backbone and add a new input embedding named **XPath embedding** to the original embedding layer. The overview structures of MarkupLM and the newly-proposed XPath Embedding are shown in Figure 2 and 4.

XPath Embedding For the i-th input token x_i, we take its corresponding XPath expression and split it by "/" to get the node information at each level of the XPath as a list, $xp_i = [(t^i_0, s^i_0), (t^i_1, s^i_1), \cdots, (t^i_d, s^i_d)]$, where d is the depth of this XPath and (t^i_j, s^i_j) denotes the tag name and the subscript of the XPath unit on level j for x_i. Note that for units with no subscripts, we assign 0 to s^i_j. To facilitate further processing, we do truncation and padding on xp_i to unify their lengths as L.

The process of converting XPath expression into XPath embedding is shown in Figure 4. For (t^i_j, s^i_j), we input this pair into the j-th tag unit embedding table and j-th subscript unit embedding table respectively, and they are added up to get the j-th unit embedding ue^i_j. We set the dimensions of these two embeddings as d_u.

$$ue^i_j = \texttt{TagUnitEmb}_j(t^i_j) + \texttt{SubsUnitEmb}_j(s^i_j)$$

We concatenate all the unit embeddings to get the intermediate representation r_i of the complete XPath for x_i.

$$r_i = [ue^i_0; ue^i_1; \cdots; ue^i_L]$$

[1] https://en.wikipedia.org/wiki/Document_Object_Model

[2] https://en.wikipedia.org/wiki/XPath

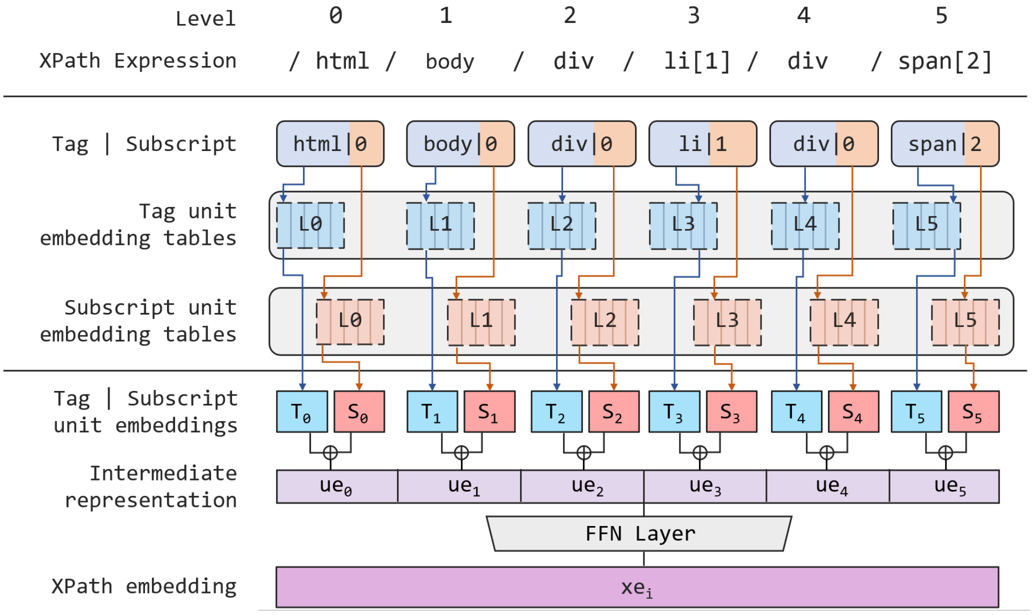


Figure 4: Overview of the XPath embedding from an XPath expression.

Finally, to match the dimension of other embeddings, we feed the intermediate representation r_i into an FFN layer to get the final XPath embedding xe_i.

$$
\begin{gathered}
xe_i = W_2[\mathrm{ReLU}(W_1 r_i + b_1)] + b_2, \\
W_1 \in \mathbb{R}^{4d_h \times Ld_u}, b_1 \in \mathbb{R}^{4d_h}, \\
W_2 \in \mathbb{R}^{d_h \times 4d_h}, b_2 \in \mathbb{R}^{d_h}
\end{gathered}
$$

where d_h is the hidden size of MarkupLM. To simplify the converting process, we have also tried replacing the FFN layer with a single linear transformation. However, this tiny modification makes the training process much more unstable and slightly hurts the performance so we keep the original design.

2.3 Pre-training Objectives

To efficiently capture the complex structures of markup-language-based documents, we propose pre-training objectives on three different levels, including token-level (MMLM), node-level (NRP), and page-level (TPM).

Masked Markup Language Modeling Inspired by the previous works (Devlin et al., 2019; Xu et al., 2020, 2021a), we propose a token-level pre-training objective Masked Markup Language Modeling (MMLM), which is designed to enhance the language modeling ability with the markup clues. Basically, with the text and markup input sequences, we randomly select and replace some tokens with `[MASK]`, and this task requires the model to recover the masked tokens with all markup clues.

Node Relation Prediction Although the MMLM task can help the model improve the markup language modeling ability, the model is still not aware of the semantics of XPath information provided by the XPath embedding. With the naturally structural DOM tree, we propose a node-level pre-training objective Node Relation Prediction (NRP) to explicitly model the relationship between a pair of nodes. We firstly define a set of directed node relationships $R \in$ {`self`, `parent`, `child`, `sibling`, `ancestor`, `descendent`, `others`}. Then we combine each node to obtain the node pairs. For each pair of nodes, we assign the corresponding label according to the node relationship set, and the model is required to predict the assigned relationship labels with the features from the first token of each node.

Title-Page Matching Besides the fine-grained information provided by markups, the sentence-level or topic-level information can also be leveraged in markup-language-based documents. For HTML-based documents, the element `<title>` can be excellent summaries of the `<body>`, which provides a supervision for high-level semantics. To efficiently utilize this self-supervised information, we propose a page-level pre-training objective Title-Page Matching (TPM). Given the element `<body>` of a markup-based document, we randomly replace the text of element <title> and ask the model to predict if the title is replaced by using the representation of token [CLS] for binary classification.

2.4 Fine-tuning

We follow the scheme of common pre-trained language models (Devlin et al., 2019; Liu et al., 2019) and introduce the fine-tuning recipes on two downstream tasks including reading comprehension and information extraction.

For the reading comprehension task, we model it as an extractive QA task. The question and context are concatenated together as the input sequence, and slicing is required when its length exceeds a threshold. For tokens of questions, the corresponding XPath embeddings are the same as [PAD] token. We input the last hidden state of each token to a binary linear classification layer to get two scores for start and end positions, and make span predictions with these scores following the common practice in SQuAD (Rajpurkar et al., 2016).

For the information extraction task, we model it as a token classification task. We input the last hidden state of each token to a linear classification layer, which has $n + 1$ categories, where n is the number of attributes we need to extract and the extra category is for tokens that belong to none of these attributes.

3 Experiments

In this work, we apply our MarkupLM framework to HTML-based webpages, which is one of the most common markup language scenarios. Equipped with the existing webpage datasets Common Crawl (CC)[3], we pre-train MarkupLM with large-scale unlabeled HTML data and evaluate the pre-trained models on web-based structural reading comprehension and information extraction tasks.

3.1 Data

Common Crawl The Common Crawl (CC) dataset contains petabytes of webpages in the form of raw web page data, metadata extracts, and text extracts. We choose one of its snapshots[4], and use the pre-trained language detection model from fasttext (Joulin et al., 2017) to filter out non-English pages. Specifically, we only take the page when the model predicts it as English with the classifier score > 0.6 and discard all the others. Besides, we only keep the tags that may contain texts (*e.g.* <div>, <span>, <li>, <a>, etc.) and delete those with no texts (*e.g.,* <script>, <style>, etc.) in these pages to save storage space. After pre-processing, a subset of CC with 24M English webpages is extracted as our pre-training data for MarkupLM.

WebSRC The Web-based Structural Reading Comprehension (WebSRC) dataset (Chen et al., 2021) consists of 440K question-answer pairs, which are collected from 6.5K web pages with corresponding HTML source code, screenshots, and metadata. Each question in WebSRC requires a certain structural understanding of a webpage to answer, and the answer is either a text span on the web page or yes/no. After adding the additional yes/no tokens to the text input, WebSRC can be modeled as a typical extractive reading comprehension task. Following the original paper (Chen et al., 2021), we choose evaluation metrics for this dataset as **Exact match (EM)**, **F1 score (F1)**, and **Path overlap score (POS)**. We use the official split to get the training and development set. Note that the authors of WebSRC did not release their testing set, so all our results are obtained from the development set.

SWDE The Structured Web Data Extraction (SWDE) dataset (Hao et al., 2011) is a real-world webpage collection for automatic extraction of structured data from the Web. It involves 8 verticals, 80 websites (10 per vertical), and 124,291 webpages (200 - 2,000 per website) in total. The task is to extract the values corresponding to a set of given attributes (depending on which vertical the webpage belongs to) from a webpage, like value for *author* in *book* pages. Following previous works (Hao et al., 2011; Lin et al., 2020; Zhou et al., 2021), we choose **page-level F1 scores** as our evaluation metrics for this dataset.

Since there is no official train-test split, we follow previous works (Hao et al., 2011; Lin et al., 2020; Zhou et al., 2021) to do training and evaluation on each vertical (*i.e.*, category of websites) independently. In each vertical, we select k consecutive seed websites as the training data and use the remaining $10 - k$ websites as the testing set. Note that in this few-shot extraction task, none of the pages in the $10 - k$ websites have been visited in the training phase. This setting is abstracted from the real application scenario where only a small

[3] https://commoncrawl.org/

[4] https://commoncrawl.org/2021/08/july-august-2021-crawl-archive-available/

Model	Modality	EM	F1	POS
T-PLM (BERT$_{\text{BASE}}$)	Text	52.12	61.57	79.74
H-PLM (BERT$_{\text{BASE}}$)	Text + HTML	61.51	67.04	82.97
V-PLM (BERT$_{\text{BASE}}$)	Text + HTML + Image	62.07	66.66	83.64
T-PLM (RoBERTa$_{\text{BASE}}$)	Text	52.32	63.19	80.93
H-PLM (RoBERTa$_{\text{BASE}}$)	Text + HTML	62.77	68.19	83.13
MarkupLM$_{\text{BASE}}$	Text + HTML	**68.39**	**74.47**	**87.93**
T-PLM (ELECTRA$_{\text{LARGE}}$)	Text	61.67	69.85	84.15
H-PLM (ELECTRA$_{\text{LARGE}}$)	Text + HTML	70.12	74.14	86.33
V-PLM (ELECTRA$_{\text{LARGE}}$)	Text + HTML + Image	73.22	76.16	87.06
T-PLM (RoBERTa$_{\text{LARGE}}$)	Text	58.50	70.13	83.31
H-PLM (RoBERTa$_{\text{LARGE}}$)	Text + HTML	69.57	74.13	85.93
MarkupLM$_{\text{LARGE}}$	Text + HTML	**74.43**	**80.54**	**90.15**

Table 1: Evaluation results on the WebSRC development set. Results on BERT and ELECTRA are obtained from the original paper (Chen et al., 2021), while those on RoBERTa are our re-running.

Model \ #Seed Sites	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
`SSM` (Carlson and Schafer, 2008)	63.00	64.50	69.20	71.90	74.10
`Render-Full` (Hao et al., 2011)	84.30	86.00	86.80	88.40	88.60
`FreeDOM-NL` (Lin et al., 2020)	72.52	81.33	86.44	88.55	90.28
`FreeDOM-Full` (Lin et al., 2020)	82.32	86.36	90.49	91.29	92.56
`SimpDOM` (Zhou et al., 2021)	83.06	88.96	91.63	92.84	93.75
MarkupLM$_{\text{BASE}}$	82.11	91.29	94.42	95.31	95.89
MarkupLM$_{\text{LARGE}}$	**85.71**	**93.57**	**96.12**	**96.71**	**97.37**

Table 2: Comparing the extraction performance (F1 score) of five baseline models to our method MarkupLM using different numbers of seed sites $k = \{1, 2, 3, 4, 5\}$ on the SWDE dataset, the results are from (Zhou et al., 2021). Each value in the table is computed from the average over 8 verticals and 10 permutations of seed websites per vertical (80 experiments in total).

set of labeled data is provided for specific websites and we aim to infer the attributes on a much larger unseen website set. The final results are obtained by taking the average of all 8 verticals and all 10 permutations of seed websites per vertical, leading to 80 individual experiments for each k. For the pre- and post-processing of data, we follow Zhou et al. (2021) to make a fair comparison.

3.2 Settings

Pre-training The size of the selected tags and subscripts in XPath embedding are 216 and 1,001 respectively, the max depth of XPath expression (L) is 50, and the dimension for the tag-unit and subscript-unit embedding (d_u) is 32. The token-masked probability in MMLM and title-replaced probability in TPM are both 15%, and we do not mask the tokens in the input sequence corresponding to the webpage titles. The max number of selected node pairs is 1,000 in NRP for each sample, and we limit the ratio of pairs with `non-others` (*i.e.*, `self, parent,` $\cdots$) labels as 80% to make a balance. We initialize MarkupLM from RoBERTa and train it for 300K steps on 8 NVIDIA A100 GPUs. We set the total batch size as 256, the learning rate as 5e-5, and the warmup ratio as 0.06. The selected optimizer is AdamW (Loshchilov and Hutter, 2019), 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`, `gradient-checkpointing` (Chen et al., 2016), and `deepspeed` (Rasley et al., 2020) to reduce GPU memory consumption and accelerate training.

Ver. \ #Seed	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
auto	70.63	89.08	94.73	95.45	98.15
book	81.89	87.43	89.40	90.26	90.35
camera	84.65	92.72	94.63	95.16	94.99
job	76.86	86.19	90.02	90.99	92.34
movie	90.53	94.87	97.85	98.91	99.37
nbaplayer	85.92	91.97	94.31	94.15	96.07
restaurant	82.76	92.25	95.87	98.70	97.04
university	83.67	95.80	98.55	98.82	98.77
Average	82.11	91.29	94.42	95.31	95.89

Ver. \ #Seed	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
auto	74.77	86.88	96.22	96.46	99.19
book	85.73	92.01	92.97	93.29	93.46
camera	85.18	95.09	96.22	96.69	96.27
job	80.64	90.67	90.41	90.72	92.99
movie	94.27	98.55	99.23	99.66	99.58
nbaplayer	88.95	94.27	97.76	98.26	98.77
restaurant	87.06	94.37	98.06	98.7	98.83
university	89.10	96.69	98.07	99.87	99.88
Average	85.71	93.57	96.12	96.71	97.37

Table 3: Evaluation results of MarkupLM (BASE on left and LARGE on right) on the SWDE dataset with different numbers of seed sites $k = \{1, 2, 3, 4, 5\}$ for training, Ver. stands for vertical while #Seed is the number of seed sites.

Fine-tuning For WebSRC, we fine-tune MarkupLM for 5 epochs with the total batch size of 64, the learning rate of 1e-5, and the warmup ratio of 0.1. For SWDE, we fine-tune MarkupLM with 10 epochs, the total batch size of 64, the learning rate of 2e-5, and the warmup ratio of 0.1. The max sequence length is set as 384 in both tasks, and we keep other hyper-parameters as default.

3.3 Results

The results for WebSRC are shown in Table 1. Selected baselines are T-PLM, H-PLM, and V-PLM in Chen et al. (2021), referring to the paper for more details. To make a fair comparison, we re-run the released baseline experiments with RoBERTa. We observe MarkupLM significantly surpass H-PLM which uses the same modality of information. This strongly indicates that MarkupLM makes better use of the XPath features with the specially designed embedding layer and pre-training objectives compared with merely adding more tag tokens into the input sequence as in H-PLM. Besides, MarkupLM also achieves a higher score than the previous state-of-the-art V-PLM model that requires a huge amount of external resources to render the HTML source codes and uses additional vision features from Faster R-CNN (Ren et al., 2015), showing that our render-free MarkupLM is more lightweight and can learn the structural information better even without any visual information. It is also worth noting that adding HTML tags as input tokens in H-PLM and V-PLM drastically increases the length of input strings, so more slicing operations are required to fit the length limitation of language models, which results in more training samples (∼860k) and longer training time, while MarkupLM does not suffer from this (only ∼470k training samples) and can greatly reduce training time.

The results for SWDE are in Table 2 and 3. It is observed that our MarkupLM also substantially outperforms the strong baselines. Different from the previous state-of-the-art model SimpDOM which explicitly sends the relationship between DOM tree nodes into their model and adds huge amounts of extra discrete features (*e.g.*, whether a node contains numbers or dates), MarkupLM is much simpler and is free from time-consuming additional webpage annotations. We also report detailed statistics with regard to different verticals in Table 3. With the growth of k, MarkupLM gets more webpages as the training set, so there is a clear ascending trend reflected by the scores. We also see the variance among different verticals since the number and type of pages are not the same.

3.4 Ablation Study

To investigate how each pre-training objective contributes to MarkupLM, we conduct an ablation study on WebSRC with a smaller training set containing 1M webpages. The model we initialized from is BERT-base-uncased in this sub-section with all the other settings unchanged. The results are in Table 4. According to the four results in #1, we see both of the newly-proposed training objectives improve the model performance substantially, and the proposed TPM (+4.6%EM) benefits the model more than NRP (+2.4%EM). Using both objectives together is more effective than using either one alone, leading to an increase of 5.3% on EM. We can also see a performance improvement (+1.9%EM) from #1d to #2a when replacing BERT with a stronger initial model RoBERTa. Finally, we get the best model with all three objectives and better initialization on larger data, as the comparison between #2a and #2b.

		Pre-training Data	Objectives			Metrics		
#	Initialization	Samples	MMLM	NRP	TPM	EM	F1	POS
1a	$\text{BERT}_{\text{BASE}}$	1M	✓			54.29	61.47	82.03
1b	$\text{BERT}_{\text{BASE}}$	1M	✓	✓		56.72	65.07	83.02
1c	$\text{BERT}_{\text{BASE}}$	1M	✓		✓	58.87	66.74	83.85
1d	$\text{BERT}_{\text{BASE}}$	1M	✓	✓	✓	59.56	68.12	84.80
2a	$\text{RoBERTa}_{\text{BASE}}$	1M	✓	✓	✓	61.48	69.15	84.32
2b	$\text{RoBERTa}_{\text{BASE}}$	24M	✓	✓	✓	68.39	74.47	87.93

Table 4: Ablation study on the WebSRC dataset, where EM, F1 and POS scores on the development set are reported. "MMLM", "NRP" and "TPM“ stand for Masked Markup Language Model, Node Relation Prediction and Title Page Matching respectively. All these models, except #2b, are pre-trained with 200k steps and the same hyper-parameter settings described in Section 3.2.

4 Related Work

Multimodal pre-training with text, layout, and image information has significantly advanced the research of document AI, and it has been the de facto approach in a variety of VRDU tasks. Although great progress has been achieved for the fixed-layout document understanding tasks, the existing multimodal pre-training approaches cannot be easily applied to markup-based document understanding in a straightforward way, because the layout information of markup-based documents needs to be rendered dynamically and may be different depending on software and hardware. Therefore, the markup information is vital for the document understanding. Ashby and Weir (2020) compared the Text+Tags approach with their Text-Only equivalents over five web-based NER datasets, which indicates the necessity of markup enrichment of deep language models. Lin et al. (2020) presented a novel two-stage neural approach named FreeDOM. The first stage learns a representation for each DOM node in the page by combining both the text and markup information. The second stage captures longer range distance and semantic relatedness using a relational neural network. Experiments show that FreeDOM beats the previous SOTA results without requiring features over rendered pages or expensive hand-crafted features. Zhou et al. (2021) proposed a novel transferable method SimpDOM to tackle the problem by efficiently retrieving useful context for each node by leveraging the tree structure. Xie et al. (2021) introduced a framework called WebKE that extracts knowledge triples from semi-structured webpages by extending pre-trained language models to markup language and encoding layout semantics.

However, these methods did not fully leverage the large-scale unlabeled data and self-supervised pre-training techniques to enrich the document representation learning. To the best of our knowledge, MarkupLM is the first large-scale pre-trained model that jointly learns the text and markup language in a single framework for VRDU tasks.

5 Conclusion and Future Work

In this paper, we present MarkupLM, a simple yet effective pre-training approach for text and markup language. With the Transformer architecture, MarkupLM integrates different input embeddings including text embeddings, positional embeddings, and XPath embeddings. Furthermore, we also propose new pre-training objectives that are specially designed for understanding the markup language. We evaluate the pre-trained MarkupLM model on the WebSRC and SWDE datasets. Experiments show that MarkupLM significantly outperforms several SOTA baselines in these tasks.

For future research, we will investigate the MarkupLM pre-training with more data and more computation resources, as well as the language expansion. Furthermore, we will also pre-train MarkupLM models for digital-born PDFs and Office documents that use XML DOM as the backbones. In addition, we will also explore the relationship between MarkupLM and layout-based models (like LayoutLM) to deeply understand whether these two kinds of models can be pre-trained under a unified multi-view and multi-task setting and whether the knowledge from these two kinds of models can be transferred to each other to better understand the structural information.

References

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*, pages 993–1003.

Colin Ashby and David Weir. 2020. Leveraging HTML in free text web named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 407–413, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Andrew Carlson and Charles Schafer. 2008. Bootstrapping information extraction from semi-structured web pages. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 195–210. Springer.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*.

Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. WebSRC: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. 2021. Lambert: Layout-aware language modeling for information extraction. In *International Conference on Document Analysis and Recognition*, pages 532–547. Springer.

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*.

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 *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 775–784. ACM.

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

Teakgyu Hong, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2021. BROS: A pre-trained language model for understanding texts in document.

Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, and C. V. Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 665–666, New York, NY, USA. Association for Computing Machinery.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021a. StructuralLM: Structural pre-training for form understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6309–6318, Online. Association for Computational Linguistics.

Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I. Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. 2021b. Selfdoc: Self-supervised document representation learning.

Bill Yuchen Lin, Ying Sheng, Nguyen Vo, and Sandeep Tata. 2020. Freedom: A transferable neural architecture for structured information extraction on web documents. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1092–1102. ACM.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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*, pages 2200–2209.

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*.

Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. In *International Conference on Document Analysis and Recognition*, pages 732–747. Springer.

Subhojeet Pramanik, Shashank Mujumdar, and Hima Patel. 2020. Towards a multi-modal, multi-task learning based pre-training framework for document representation learning.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Te-Lin Wu, Cheng Li, Mingyang Zhang, Tao Chen, Spurthi Amba Hombaiah, and Michael Bendersky. 2021. Lampret: Layout-aware multimodal pretraining for document understanding.

Chenhao Xie, Wenhao Huang, Jiaqing Liang, Chengsong Huang, and Yanghua Xiao. 2021. Webke: Knowledge extraction from semi-structured web with pre-trained markup language model. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2211–2220.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021a. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

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 *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021b. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding.

Yichao Zhou, Ying Sheng, Nguyen Vo, Nick Edmonds, and Sandeep Tata. 2021. Simplified dom trees for transferable attribute extraction from the web.