DocHieNet: A Large and Diverse Dataset for Document Hierarchy Parsing

Hangdi Xing[1*], **Changxu Cheng**[2*], **Feiyu Gao**[2†], **Zirui Shao**[1],
Zhi Yu[1†], **Jiajun Bu**[1], **Qi Zheng**[2], **Cong Yao**[2]
[1]Zhejiang University [2]Alibaba Group
{xinghd, shaozirui, yuzhirenzhe, bjj}@zju.edu.cn, ccx0127@gmail.com,
feiyu.gfy@alibaba-inc.com, yongqi.zq@taobao.com, yaocong2010@gmail.com

Abstract

Parsing documents from pixels, such as pictures and scanned PDFs, into hierarchical structures is extensively demanded in the daily routines of data storage, retrieval and understanding. However, previously the research on this topic has been largely hindered since most existing datasets are small-scale, or contain documents of only a single type, which are characterized by a lack of document diversity. Moreover, there is a significant discrepancy in the annotation standards across datasets. In this paper, we introduce a large and diverse document hierarchy parsing (DHP) dataset to compensate for the data scarcity and inconsistency problem. We aim to set a new standard as a more practical, long-standing benchmark. Meanwhile, we present a new DHP framework designed to grasp both fine-grained text content and coarse-grained pattern at layout element level, enhancing the capacity of pre-trained text-layout models in handling the multi-page and multi-level challenges in DHP. Through exhaustive experiments, we validate the effectiveness of our proposed dataset and method[1].

1 Introduction

Nowadays, an overwhelming amount of information is generated daily and stored in documents as pixels, such as pictures and scanned PDFs, rather than in hierarchically structured formats. It introduces a significant challenge in practice, as structured formats are essential for efficient database storage and standardized data handling (Johnson et al., 2003; Clifton and Garcia-Molina, 2000), as well as downstream tasks, such as information retrieval and natural language processing (Wilkinson, 1994; Dasigi et al., 2021). Particularly, it has been studied that documents with structural metadata further enhance the capabilities of large language models (LLMs), which has been outstanding across various domains, in processing lengthy documents and knowledge-intensive tasks (Saad-Falcon et al., 2023; Gao et al., 2023).

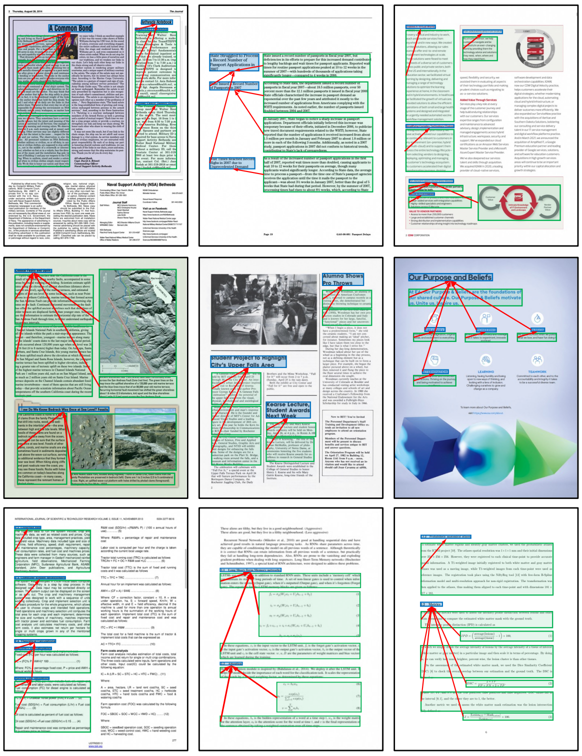


Figure 1: Examples of various page layouts and structures in DocHieNet. Blue and green boxes represent layout elements of titles and paragraphs. Red lines refer to the hierarchical relations. Only part of the hierarchical relations are shown for clarity.

Document hierarchy parsing (DHP) aims at reconstructing the hierarchical relationships among document layout elements (e.g., titles, paragraphs, figures), as shown in Fig. 1 and thus organizing the document in a machine-understandable, hierarchically structured format. For documents as pixels, the layout elements can be extracted by

[*] Equal contribution.

[†] Corresponding author.

[1]The dataset and code are available at https://github.com/AlibabaResearch/AdvancedLiterateMachinery/tree/main/DocumentUnderstanding/DocHieNet

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 1129–1142
November 12-16, 2024 ©2024 Association for Computational Linguistics

off-the-shelf document layout analysis systems (Zhong et al., 2019b), and the DHP model focuses on predicting the hierarchical relationship among them. Issues on previous datasets have hindered the progress of research and application. First, the datasets struggle to reflect the complexity of real-world documents. The arXivdocs (Rausch et al., 2021) and E-Periodica (Rausch et al., 2023) are considered small-scale, containing only hundreds of single pages. Regarding HRDoc and Comp-HRDoc (Ma et al., 2023; Wang et al., 2024), although they are large-scale and exhibit various lengths, they contain only monotonous scientific articles, which share similar layout designs and hierarchical structures, such as examples in the 3rd row of Fig. 1. Second, the annotation standards are inconsistent. For instance, the granularity of layout element annotations varies among datasets, including those based on text line level and layout block level. Moreover, their definitions of hierarchical relations also differ with the varying definitions of layout elements.

Regarding the models, DHP presents two primary challenges: the handling of extended, multi-page inputs and the comprehension of both textual content and the high-level layout relationships. Previous works employ heuristic rules (Rausch et al., 2021) and LSTM networks (Rausch et al., 2023) for their efficiency with lengthy inputs. Ma et al. (2023) utilize a pre-trained language model (PLM) as the encoder to enhance the model performance. But this model extracts the text features of each layout element independently, thus overlooking the fine-grained contexts of layout elements.

As a result of the issues with the dataset and model design, existing DHP methods struggle to be applicable in the real-world scenarios. In order to promote the development of DHP in more complex and realistic scenarios, we proposed DocHieNet, a large-scale, multi-page, multi-domain, multi-layout and bi-lingual dataset for DHP. DocHieNet contains 1673 multi-page documents from different scenarios including public sector, research, industry, etc. The multi-page documents, up to 50 pages, are characterized by large heterogeneity in their presentation and thus complex document structures (Fig. 1), which are close to real-world conditions. The data collection of DocHieNet inherently encourages the development of models capable of addressing DHP on highly diverse documents. Statistics of the datasets are summarized in Tab. 1.

With DocHieNet available, we propose a transformer-based framework, DHFormer, which effectively overcomes the multi-page and multi-level challenges in DHP. It adopts a sparse text-layout encoder, derived from the powerful layout-aware language models (LMs) (Xu et al., 2021; Luo et al., 2023) to represent the layout elements with enriched fine-grained contexts. Subsequently, a layout element-level reasoning decoder is exploited to capture collective information from multiple pages at the global range. Besides, DHFormer leverages the page embeddings and inner-layout position embeddings in order to better depict the cross-page and multi-level patterns. Experiments show that the proposed method is highly competitive and outperforms previous methods by a large margin.

Our main contributions can be summarized as follows:

- We have created DocHieNet, a novel large-scale, multi-page, multi-domain and multi-layout dataset for facilitating the development of generic DHP models.
- We propose DHFormer, which effectively enhances text-layout models to better grasp both text content and coarse-grained patterns between layout elements in multi-page and multi-level DHP scenarios.
- Statistical and experimental results validate the challenging nature of the proposed DocHieNet dataset and the effectiveness of the DHFormer method. The dataset and model are publicly available.

2 Related Work

2.1 Document AI

Document AI involves automated reading, understanding and extracting information from visually-rich documents (VRDs) (Liu et al., 2019; Li et al., 2020a; Cui et al., 2021; Xing et al., 2023; Shao et al., 2023). As the world is going digital, it has received a heightened focus on its impact and significance. The Document Layout Analysis (DLA) task (Namboodiri and Jain, 2007), which refers to the detection and recognition of layout elements such as text and table/figure region, has seen a surge of research achievements (Li et al., 2020b; Pfitzmann et al., 2022). Based on these works, datasets and methods are proposed to further understand the semantic relationships of layout elements and extract their hierarchical structure (Rausch et al., 2021,

Dataset	#Docs	#Pages	#M.P.	C.P.R&S	A.M.	Document Type	Language
arXivdocs	362	362	1	(0%, 0)	Manual	Scientific papers	En
HRDoc	2500	31651	35	(24.9%, 2.4)	Automatic	Scientific papers	En
E-Periodica	542	542	1	(0%, 0)	Manual	Magazines	En, DE, FR, IT
DocHieNet	1673	15610	50	(37.4%, 5.4)	Manual	Multiple Types	En, Zh

Table 1: Statistics of Document Hierarchy Parsing Datasets. M.P. and A.M. denote the max pages and annotation means respectively. C.P.R.& S. stands for the cross-page ratio and span, which consists of the macro-average of the proportion and max page span of the cross-page hierarchical relations.

2023), i.e. document hierarchy parsing, which plays an indispensable role in document AI.

2.2 Document Hierarchy Parsing

There are a handful number of datasets available for DHP. Rausch et al. (2021) are the forerunners for contributing the arXivdocs, which contains only 362 single pages randomly selected from arXiv. Ma et al. (2023) propose the HRDoc dataset with 2500 multi-page documents from ACL/arXiv and Wang et al. (2024) improve the labels. Nevertheless, they are limited to scientific articles, which share similar structures. Rausch et al. (2023) mitigate this homogeneity by introducing the E-Periodica, which is comprised of 542 single pages from magazines. However, E-Periodica still exhibits issues of limited pagination and small scale.

The DHP model requires accommodating long document inputs, which has led prior models (Rausch et al., 2021, 2023) to rely on heuristic rules or LSTM networks (Hochreiter and Schmidhuber, 1997), for their reduced computational complexity. In order to improve the performance, Ma et al. (2023) employ a PLM to independently encode each layout element. But the model fails to address the multi-level challenge in DHP by overlooking the fine-grained contexts of layout elements.

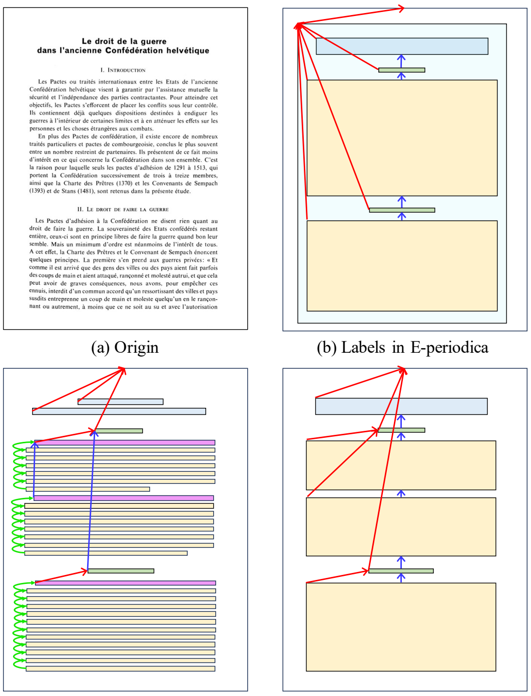
2.3 Long-document Transformers

Transformers (Vaswani et al., 2017) have become the fundamental model for natural language processing tasks, which requires quadratic space dependency. Early works such as (Beltagy et al., 2020) propose types of sparse attention to tackle this challenge. Nonetheless, such approaches demand additional pre-training. Ivgi et al. (2022); Xie et al. (2023) show that building a sparse transformer via document chunking, while keeping the attention pattern unchanged, forgoes the extra pre-training and effectively handles lengthy texts. Since the long multi-page VRDs lack pre-training corpora, Tito et al. (2022); Kang et al. (2024) follow the chunk-based method to solve the multi-page document VQA. However, their page-level design cannot be directly implemented on DHP which focuses on finer-grained relationships among layout elements.

3 Problem Definition

In this paper, we consider the DHP as recognizing the hierarchical structure among layout elements. Specifically, the input is given as a multi-page document along with M extracted layout elements $E = \{E_1, E_2, ..., E_M\}$ in traversal order, which can be obtained by the off-the-shelf optical character recognition (OCR) and document layout analysis system (Cheng et al., 2023). The output is the hierarchical structure of the elements (E, R), where R is the relation set which captures relationships between layout elements. Relation R_j is defined as a tuple (E_{parent}, E_{child}) which represents a hierarchical relation between elements.

The definitions of the layout elements and their relationships vary among datasets. Fig. 2 depicts a document image, with annotations visualized according to labeling systems of different datasets. E-Periodica (See Fig. 2 (b)), defines layout elements as multi-granular content blocks with hierarchical relations which exist between elements of different granularities, and sequential relations which indicate reading order. This setup imposes stringent requisites on the layout analysis module for multi-granular elements, and it also results in semantically incomplete elements by annotating single pages separately. In HRDoc, annotations are based on text lines, simplifying issues of multi-granularity by requiring the model to additionally identify text lines belonging to the same layout block (See green lines of ‘connect’ relationship in Fig. 2 (c)). This approach neglects the advanced document layout analysis models. Besides, the



Le droit de la guerre
dans l'ancienne Confédération helvétique

I. Introduction

Les Pactes ou traités internationaux entre les Etats de l'ancienne Confédération helvétique visent à garantir par l'assistance mutuelle la sécurité et l'indépendance des parties contractantes. Pour atteindre cet objectifs, les Pactes s'efforcent de placer les conflits sous leur contrôle. Ils contiennent déjà quelques dispositions destinées à endiguer les guerres à l'intérieur de certaines limites et à en atténuer les effets sur les personnes et les choses étrangères aux combats.

En plus des Pactes de confédération, il existe encore de nombreux traités particuliers et pactes de combourgeoisie, conclus le plus souvent entre un nombre restreint de partenaires. Ils présentent de ce fait moins d'intérêt en ce qui concerne la Confédération dans son ensemble. C'est la raison pour laquelle seuls les pactes d'adhésion de 1291 à 1513, qui portent la Confédération successivement de trois à treize membres, ainsi que la Charte des Prêtres (1370) et les Convenants de Sempach (1393) et de Stans (1481), sont retenus dans la présente étude.

II. Le droit de faire la guerre

Les Pactes d'adhésion à la Confédération ne disent rien quant au droit de faire la guerre. La souveraineté des Etats confédérés restant entière, ceux-ci sont en principe libres de faire la guerre quand bon leur semble. Mais un minimum d'ordre est néanmoins de l'intérêt de tous. A cet effet, la Charte des Prêtres et le Convenant de Sempach énoncent quelques principes. La première s'en prend aux guerres privées: « Et comme il est arrivé que des gens des villes ou des pays aient fait parfois des coups de main et aient attaqué, rançonné et molesté autrui, et que cela peut avoir de graves conséquences, nous avons, pour empêcher ces ennuis, interdit d'un commun accord qu'un ressortissant des villes et pays susdits entreprenne un coup de main et moleste quelqu'un en le rançonnant ou autrement, à moins que ce ne soit au su et avec l'autorisation

(a) Origin

(b) Labels in E-periodica

(c) Labels in HRDoc

(d) Labels in DocHieNet

Figure 2: Illustration of the label systems in different datasets. Red and blue lines denote 'hierarchical' and 'sequential' relationships, and green lines indicate 'connect' relationships. The point at the top of the document represents the root of document.

prevalence of the 'connect' relationship far exceeds other relations, making line-level evaluation a poor reflection of prediction quality due to the simplicity of the 'connect' pattern compared to the more complex hierarchical relationship.

Integrating the merits of different definitions and referencing prevailing works in the document layout analysis, we design the labeling system of DocHieNet to annotate only fine-grained layout blocks and capture both hierarchical and sequential relationships, as illustrated in Fig. 2 (d).

4 Dataset

The DocHieNet contains a total of 1673 documents, of which 1110 are in English and 563 are in Chinese. It covers a wide range of domains including legal, financial, educational, technical, and scientific documents. Furthermore, as illustrated in Fig. 1, the documents are of diversified layout.

4.1 Document Collection

The documents of the DocHieNet dataset are selected from diverse data sources including comprehensive document VQA datasets (Tito et al., 2022; Landeghem et al., 2023), government public release, data directory services for financial reports and other aggregate websites. Information on the search procedure and resources of data is distributed as a part of the DocHieNet dataset. We manually select representative documents of their type while preventing too many samples gathered in a single type. Extra caution is exercised in ensuring that all samples are free to use and eliminating samples that could potentially raise complications pertaining to privacy considerations.

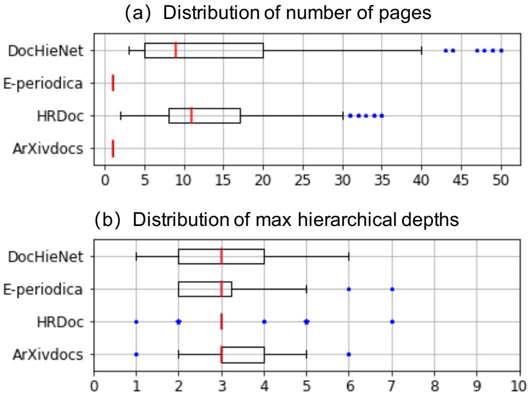


Figure 3: Distribution of number of pages and max hierarchical depths of the four datasets shown in Tab. 1.

4.2 Annotation Process

The campaign begins with annotating layout elements. Based on the observation of common layout features in the collected data and previous definitions of layout element classes, we define a taxonomy of 19 types: {*title, sub-title, section-title, text, formula, TOC-title, TOC, figure, fig-title, fig-caption, table, tab-title, tab-caption, header, footer, page-number, footnote, endnote, sidebar*}. The statistics of layout elements are summarized in Appendix A.1. In this phase, the layout elements are annotated with their categories, positions and text content, organized in reading order across pages.

Given the diversity in document themes and layouts, the hierarchical relationship annotation becomes complex. We thus supply precise annotation guidelines and plenty of examples for typical document types. Twelve experienced annotators undertake this task adhering strictly to these guidelines, with three specialists in the document understanding area performing three rounds of quality checks. Within our corpus, many documents are lengthy, with recurring layout patterns. To improve annotation efficiency and reduce pattern redundancy, we have truncated half of the documents (totaling 835).

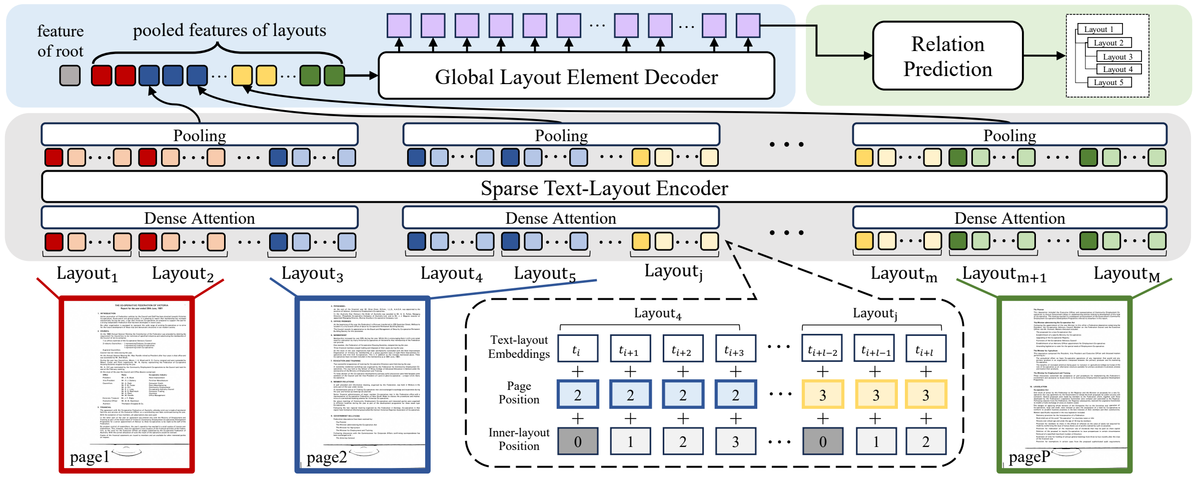


Figure 4: An overview of DHFormer. The sparse text-layout encoder efficiently enriches the input representations with fine-grained contexts. Then the decoder takes as input the pooled layout features of the document and reasons at global range. Finally the relations are predicted based on features of layout elements.

4.3 Data Split and Statistics

We carefully split the annotated documents into a train-set of 1512 documents and a test-set of 161 documents. To prevent over-fitting to a particular pattern, we regulate the balance of documents from diverse sources within the splits. Additionally, the documents in the test-set encompass fully annotated documents exclusively, and thus DocHieNet is able to gauge the generalization ability of models across documents of varying lengths. More details of the splits are summarized in Appendix A.2.

Our research entails statistical evaluations of the datasets, which reveals that DocHieNet is of higher diversity compared with previous DHP datasets. We present the principal statistical data of the dataset in Tab. 1. It is evident that DocHieNet represents the largest manually annotated dataset and is the sole dataset with multiple types of documents.

In terms of document length, as depicted in Fig. 3 (a), DocHieNet exhibits a more extensive and varied distribution of page numbers. Pertaining to the complexity of document hierarchy, DocHieNet also demonstrates significant diversity. It encompasses a larger proportion and a broader span of cross-page relationships, as summarized in Tab. 1. Furthermore, in the aspect of the depth of the document hierarchy tree, DocHieNet is also more diversified. Previous datasets, due to the homogeneity of the documents, exhibit a more concentrated distribution as shown in Fig. 3 (b).

5 Method

The proposed DHFormer framework, as illustrated in Fig. 4, leveraging both fine-grained and holistic information, and making full use of pre-trained layout-aware LMs, effectively tackles the multi-page and multi-level challenges in DHP. Firstly, the entire document, including tokens and their 2D positions, is fed into a sparse text-layout encoder $\mathbf{E}_{sp}$ to create a fine-grained contextualized representation for each token. Then, through pooling, the information is input into a layout element-level decoder $\mathbf{D}$. The decoder captures collective information from higher-level and global contexts to obtain representations of layout elements. We specially equip the text-layout model with additional page embeddings and inner-layout position embeddings to enhance the capacity of modeling cross-page and multi-level relations. Finally, the contextualized layout features are fed into the relation prediction head to get the final output.

5.1 Sparse Text-layout Encoder

Layout-aware LMs (Xu et al., 2019, 2021; Luo et al., 2023) can be taken as the text-layout encoder. In multi-page VRDs, the number of tokens N usually exceeds the input limitations l of the pre-trained encoder. There are various strategies to extend their attention mechanism to handle long inputs [2]. In this section, we employ a chunk-based sparse transformer which keeps the dense atten-

[2] Discussion on different sparse transformer strategies is provided in the experiments.

tion within chunks and thus better exploits the LMs pre-trained on single pages (Ivgi et al., 2022; Xie et al., 2023). We break down the document to K chunks $C = \{C_1, ..., C_K\}$. Each chunk contains the maximum number of layout elements such that the total number of their tokens does not exceed l. The chunks are encoded distributively, so the attention map in the encoder $\mathbf{E}_{sp}$ is factorized into dense attention only within chunks :

$$\widetilde{X} = \text{Att}(X, C) = (a(x_i, C_{k_i}))_{i \in 1,...,N} \quad (1)$$

$$a(x_i, C_{k_i}) = \text{softmax}(\frac{(W_q x_i) K_{k_i}^T}{\sqrt{d}}) V_{k_i} \quad (2)$$

Where X is the input embeddings and C_{k_i} is the chunk to which x_i belongs, and :

$$K_{k_i} = (W_k x_j)_{x_j \in C_{k_i}}, V_{k_i} = (W_v x_j)_{x_j \in C_{k_i}} \quad (3)$$

W_q, W_k, and W_v represent the weight matrices and d is the hidden size of the model.

In this way, we enrich the fine-grained contexts of tokens rather than only within layout elements, while keeping computational cost in check. The vanilla self-attention complexity of the entire document is $O(N^2)$. The attention factorized within chunks has the complexity of $O(|C_1|^2 + |C_2|^2 + ... + |C_k|^2)$. Supposing that the size of chunks are all of l for estimation, then there is $N = l \cdot K$ and the complexity of the factorized attention in the sparse text-layout encoder is $O(l \cdot N)$.

5.2 Position Embeddings

We further add two types of embeddings to the text-layout models, which are specially designed for the multi-page and multi-level settings in DHP:

Page embeddings denote the page location on which the input is located. It is computed as $e^{pg} = \text{Linear}(\text{sinPE}(pn_i))$, where pn_i is the absolute page number of ith input, sinPE is the sinusoidal positional encoding. It can connect layouts from the same page and distinguish layouts from different pages. The 2D position embeddings alone can be confusing in the multi-page scenario since layouts from different pages may overlap.

Inner-layout position embeddings are calculated by $e^{in} = \text{PosEmb1D}(rp_i)$, where rp_i is the relative position of ith input within its corresponding layout element, and PosEmb1D is the 1D position embedding function of the encoder. It helps the model obtain the awareness of the boundaries of layout elements in text sequences, which facilitates better representation of layout elements.

Formally, the ith input embedding is computed as $x_i = t_i + e_i^{pg} + e_i^{in}$, where t_i is the original text-layout embedding of the encoder.

5.3 Global Layout Element Decoder

For each layout element E_i, its representation H_i is derived by pooling the feature of its first token. An additional learnable root embedding H_0 is utilized since some layouts have the root node as the parent. The features of layouts are concatenated and passed into a transformer-based decoder $\mathbf{D}$, producing the final representations $\hat{H}_i$ of layouts as :

$$\{\hat{H}_i\}_{i=0,...,M} = \mathbf{D}(\{H_i\}_{i=0,...,M}) \quad (4)$$

This module refines the layout features at the global range and further breaks down the barriers between chunks. Considering that the number of layouts is also unlimited in real cases, shifted sparse attention (SSA) (Chen et al., 2023) is utilized to efficiently support a greater number of layout elements.

5.4 Prediction

Finally, the relations between layout elements are predicted as dependency parsing following (Luo et al., 2023), where a bilinear layer is applied:

$$p_{ij} = \text{Sigmoid}(\text{Bilinear}(\hat{H}_i, \hat{H}_j)) \quad (5)$$

Then the parent of E_i, in terms of hierarchical relationships, is predicted by $\text{argmax}(\{p_{ij}\}_{j=0,..,M})$ to obtain the relation pair. During training, the cross-entropy loss is used.

6 Experiment

6.1 Implementation Details

We employ pre-trained GeoLayoutLM (Luo et al., 2023) as the basic text-layout encoder and a 2-layer SSA with a window size of 48 as the decoder. The AdamW optimizer (Loshchilov and Hutter, 2017) is employed for training with a base learning rate of 4e-5. The training epoch is set to 100 as the default, where the learning rates progressively decrease to 1e-6. During training, we set the max tokens of the text-layout encoder as 512 with the max number of chunks, as 32 (128 for testing). All the experiments of DHFormer are performed on the platform with 2 NVIDIA Tesla V100 GPUs.

6.2 Evaluation Protocols

We employ both F1-score to measure the correctness of predicted relation triples (Rausch et al.,

Dataset	arXivdocs		HRDS		HRDH		E-Periodica		DocHieNet	
metric	F-1	TEDS	F-1	TEDS	F-1	TEDS	F-1	TEDS	F-1	TEDS
DocParser	58.14	29.11	56.84	28.71	47.36	22.39	35.20	18.67	23.31	6.81
DSPS	-	-	-	81.74	-	69.71	-	-	-	-
DOC	-	-	-	95.10	-	85.48	-	-	-	-
DSG	81.17	72.47	84.78	83.24	74.04	64.33	67.17	60.14	53.51	33.90
DHFormer	**98.45**	**95.04**	**99.34**	**98.69**	**93.40**	**89.14**	**92.53**	**84.85**	**77.82**	**57.64**

Table 2: Summary of performance of document hierarchy parsing methods across different datasets. Bold figures indicate the best results of all models.

	Anno. Format		arXivdocs		HRDS		HRDH		E-Periodica	
Settings	Train	Test	F-1	TEDS	F-1	TEDS	F-1	TEDS	F-1	TEDS
1	DHN	DHN	98.45	95.04	99.34	98.69	93.40	89.14	92.53	84.85
2	DHN	origin	-	-	99.87	99.73	98.36	97.31	-	-
3	origin	origin	99.70	97.42	99.57	97.98	96.69	92.63	95.76	93.09

Table 3: Summary of performance of DHFormer on different datasets with their original annotation formats. 'DHN' and 'origin' refer to the annotation format of DocHieNet and the original dataset respectively.

2023) and Tree-Edit-Distance based Similarity (TEDS) to assess the entire document tree structure (Zhong et al., 2019a; Hu et al., 2022). More details of evaluation are introduced in the Appendix A.3.

6.3 Comparison of Document Hierarchy Parsing Models across Datasets

We assess a group of DHP models to investigate their performance across different datasets, including DocParser (Rausch et al., 2021), DSPS (Ma et al., 2023), DOC (Wang et al., 2024) and DSG (Rausch et al., 2023). The baselines are summarized with more details in Appendix A.4. As mentioned in Sec. 3, there exists inconsistency across different datasets. To facilitate a comprehensive comparison, we map the labels of previous datasets onto the DocHieNet format. For DocParser, we do not alter the data containing multi-granularity layout elements, as its empirical rules are predicated on such annotations. Regarding the DSPS and DOC model, we refer to the reported evaluation results, specifically the evaluation conducted on the text line level. The results are in Tab. 2.

An analysis of each row reveals the notably higher complexity of DocHieNet compared to other datasets. For example, DHFormer achieves commendable results on previous datasets, but its performance on DocHieNet indicates substantial room for enhancement. A vertical comparison in each column illustrates the superiority of DHFormer. Despite DSG integration of multi-modal features, the absence of document-specific pre-training limits its effectiveness in the data-scarce scenario. Although the DSPS model employs the PLM, the layout elements are encoded separately with only limited contexts. DHFormer overcomes the drawbacks of previous model with the specially designed architecture to better exploit the pre-trained layout-aware LMs on the multi-page and multi-level DHP setting. We also investigate the performance of DHFormer on documents of different languages in Appendix A.5.

6.4 Model Performance on Different Annotation Formats

In order to provide a more comprehensive assessment of the proposed model, we evaluate the performance of DHFormer on different datasets with their original annotation formats as shown in Tab. 3. Setting 1 is the same as that in Tab. 2. In setting 2 the model is trained with labels of DocHieNet standard, while the results are transformed back into the original standards for evaluation. Note that we have manually transformed the E-Periodica and arXivdocs into DocHieNet standard, so the predicted results can not be directly transformed back. In setting 3, the model is trained and evaluated on the original annotations of the datasets.

For results on HRDoc datasets, the results in setting 2 become obviously higher than in setting 1. It

Encoder	F1	TEDS
XLM-RoBERTa	69.13	50.61
BROS	74.10	53.39
LayoutLMv3	75.83	56.40
GeoLayoutLM	77.82	57.64

Table 4: The model performance of DHFormer with different encoders.

ID	Model	Train	Eval	HRDS	HRDH
1	DSPS	Line	Line	81.74	69.71
2a	DHFormer	Line	Line	97.98	92.63
2b	DHFormer	Line	Layout	91.69	83.91
2c	DHFormer	Layout	Line	99.73	97.31
3a	DHFormer	Layout	Layout	98.69	89.14
3b	DHFormer*	Layout	Layout	94.32	86.87

Table 5: Experiment results on HRDoc with different annotation granularity. DHFormer* refers to the end-to-end results with a layout analysis system.

is because the backward transformation splits the layout element into text lines and adds 'connect' relations among them, which are exactly ground-truth relations. For E-Periodica and arXivdocs datasets, the performance in setting 3 is higher, mainly because the layout information provides strong clues for the relationships defined in these datasets. In setting 3, directly training and testing the model on the original datasets also shows commendable results, which indicates the effectiveness and flexibility of DHFormer.

6.5 Model Performance with Different Pre-trained Encoders

We conduct additional experiments by replacing GeoLayoutLM in the encoder with other representative layout-aware LMs, including BROS (Hong et al., 2022) and LayoutLMv3 (Huang et al., 2022) along with a plain-text LM XLM-RoBERTa (Conneau et al., 2019) of equal parameter size. The results are summarized in Tab. 4. It shows that the performance fluctuates slightly according to different pre-trained models, while consistently outperforming previous methods. It demonstrates the flexibility and robustness of the framework.

6.6 Discussion on Paradigms of Annotations

In this section, we conduct an analysis of different annotation paradigms through statistical data

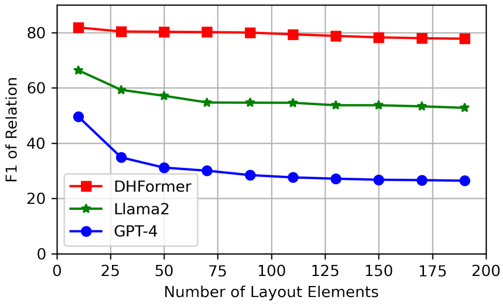


Figure 5: Comparison of the DHFormer and LLMs, in terms of model performance in relation to variations in document length.

and experimental results. As mentioned in Sec. 3, the layout element defined in E-Periodica is solely applicable to single-page documents. It fails to encompass cross-page relationships, which constitute a significant proportion in multi-page documents, as summarized in Tab. 1. The limitations of this annotation paradigm are self-evident.

The HRDoc annotation system, by establishing relations among text lines, integrates the tasks of layout analysis and hierarchy parsing. Experiment results indicate that this setting is not as ideal as it appears. We train DHFormer with the original HRDoc annotations and conducted evaluations on both text line (2a), and layout block level (2b) by merging lines into blocks according to the predictions. We also break down the results of DHFormer trained with block-level annotations into text lines to make a thorough comparison (2c). The evaluation results based on layout blocks are significantly lower, which indicates that text line-level evaluations inadequately reflect the actual quality of the predicted hierarchy as mentioned in Sec. 3.

We further compare the end-to-end inference outcomes based on layout blocks detected by a layout analysis system using CenterNet (Zhou et al., 2019). Employing the results of the layout analysis model as input demonstrated a decline (from 3a to 3b), albeit still surpassing the outcomes of line-level prediction after merging text lines into layout blocks for evaluation (2b), which further indicates the merit of the annotation paradigm of DocHieNet.

6.7 Discussion on Large Language Models

Recently, large language models have been gaining adoption in different domains and accommodate more extensive text inputs, such as 128K tokens. The GPT-4 represents one of the state-of-the-art

ID	STS	WinS	F-1	TEDS
a	chunk	layout	62.41	46.75
b	chunk	page	75.66	55.07
c	stride	512	73.98	54.38
d	chunk	512	77.82	57.64

Table 6: The comparison of different sparse transformer strategies (STSs) and window size (WinS).

ID	PageE.	InnerE.	F-1	TEDS
a	w/o	w/o	73.66	52.54
b	w	w/o	75.77	55.14
c	w/o	w	75.14	54.41
d	w	w	77.82	57.64

Table 7: Ablations of the page embeddings and inner-layout position embeddings.

LLMs and Llama2 is a prevalent open-source large model in academia. We take them as baselines to evaluate LLMs on DocHieNet. The prompt for GPT-4 employs in-context learning (ICL) (Brown et al., 2020) , while Llama2 is fine-tuned on our dataset. Further details of the APIs, prompt and fine-tuning process are provided in Appendix A.6.

The comparison in terms of relation F-1 is shown in Fig. 5. As illustrated, DHFormer outperforms GPT-4 based on ICL or fine-tuned Llama2. Moreover, with the increment in the length of the documents evaluated, DHFormer only exhibits a slight decline. This can be attributed to its adeptly balancing detailed and holistic information, enhancing its overall performance. Besides, the decoder reasons at above-token level with collective information, which prevents the model from being overwhelmed by excessive details and consequently bolsters the model on lengthy documents.

6.8 Ablations of Design Choices

First, we assess the impact of different sparse transformer strategies (STS). We conducted experiments with chunks of varying sizes, and implemented a sliding window attention mechanism (Beltagy et al., 2020) with the same initialization. Chunking at the layout level evidently suffers from inadequate context according to the comparison of Tab. 6 (a) and Tab. 6 (d). Chunking at the page level, as shown in Tab. 6 (b), also leads to slight information loss due to the frequent cross-page relationships among layout elements. Employing the sliding window obviates the need for chunking. However, it modifies the attention pattern, and thus often necessitates further pre-training (Ivgi et al., 2022). In the scenario of multi-page long VRDs with a scarcity of pre-training data, the chunk-based method shows its superiority, which is indicated by the difference between Tab. 6 (c) and Tab. 6 (d).

Then we evaluate the effectiveness of the page embeddings and inner-layout position embeddings in Tab. 7. Results indicate that a performance boost can be achieved by adding one type of embedding respectively, while the concurrent use of both embeddings results in the best model performance.

7 Conclusion

In this paper, we present DocHieNet, a DHP dataset featuring large-scale, multi-page, multi-domain, multi-layout and bi-lingual documents. We carry out detailed analyses of data statistics, annotation paradigms and evaluation using various baselines. Our findings demonstrate the challenging nature of the DocHieNet and the advantage of its annotations format. Furthermore, we introduce an effective framework, DHFormer, which consistently improves the model performance, particularly on the complex DocHieNet dataset. We hope this work could not only advance the understanding of DHP task but also set a foundation for future exploration.

Limitations

Despite the significant effectiveness that our proposed dataset DocHieNet and method DHFormer represent, we acknowledge the limitations that while the dataset includes a vast array of document types and layouts, it may not encompass all possible variations seen in the wild. Future work could expand the dataset to include even more diverse and challenging documents, ensuring that models are more robust against more types of documents encountered in the real-world applications.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62372408) and the National Key R&D Program of China (No. 2021YFB2701100).

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *ArXiv*, abs/2309.12307.

Hiuyi Cheng, Peiyu Zhang, Sihang Wu, Jiaxin Zhang, Qi Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. 2023. M6doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15138–15147.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Chris Clifton and Hector Garcia-Molina. 2000. The design of a document database. *Proceedings of the ACM conference on Document processing systems*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Annual Meeting of the Association for Computational Linguistics*.

Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *ArXiv*, abs/2111.08609.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *AAAI*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Jun Du, and Jiajia Wu. 2022. Multimodal tree decoder for table of contents extraction in document images. *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1756–1762.

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 *ACM Multimedia*.

Maor Ivgi, Uri Shaham, and Jonathan Berant. 2022. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 11:284–299.

Stephen B. Johnson, David A. Campbell, M. Krauthammer, P. Karina Tulipano, Eneida A. Mendonça, Carol Friedman, and George Hripcsak. 2003. A native xml database design for clinical document research. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, page 883.

Lei Kang, Rubèn Pérez Tito, Ernest Valveny, and Dimosthenis Karatzas. 2024. Multi-page document visual question answering using self-attention scoring mechanism. *ArXiv*, abs/2404.19024.

Jordy Van Landeghem, Rubèn Pérez Tito, Łukasz Borchmann, Michal Pietruszka, Pawel J'oziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Ackaert, Ernest Valveny, Matthew B. Blaschko, Sien Moens, and Tomasz Stanislawek. 2023. Document understanding dataset and evaluation (dude). *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19471–19483.

Liangcheng Li, Feiyu Gao, Jiajun Bu, Yongpan Wang, Zhi Yu, and Qi Zheng. 2020a. An end-to-end ocr text re-organization sequence learning for rich-text detail image comprehension. In *European Conference on Computer Vision*.

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

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 32–39, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. 2023. Geolayoutlm: Geometric pre-training for visual information extraction. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7092–7101.

Jiefeng Ma, Jun Du, Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Huihui Zhu, and Cong Liu. 2023. Hrdoc: dataset and baseline method toward hierarchical reconstruction of document structures. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Anoop M. Namboodiri and Anil K. Jain. 2007. Document structure and layout analysis.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed Samy Nassar, and Peter W. J. Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. 2021. Docparser: Hierarchical document structure parsing from renderings. In *AAAI Conference on Artificial Intelligence*.

Johannes Rausch, Gentiana Rashiti, Maxim Gusev, Ce Zhang, and Stefan Feuerriegel. 2023. Dsg: An end-to-end document structure generator. *ArXiv*, abs/2310.09118.

Jon Saad-Falcon, Joe Barrow, Alexa F. Siu, Ani Nenkova, Ryan Rossi, and Franck Dernoncourt. 2023. Pdftriage: Question answering over long, structured documents. *ArXiv*, abs/2309.08872.

Zirui Shao, Feiyu Gao, Zhongda Qi, Hangdi Xing, Jiajun Bu, Zhi Yu, Qi Zheng, and Xiaozhong Liu. 2023. GEM: Gestalt enhanced markup language model for web understanding via render tree. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Rubèn Pérez Tito, Dimosthenis Karatzas, and Ernest Valveny. 2022. Hierarchical multimodal transformers for multi-page docvqa. *ArXiv*, abs/2212.05935.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Jiawei Wang, Kai Hu, Zhuoyao Zhong, Lei Sun, and Qiang Huo. 2024. Detect-order-construct: A tree construction based approach for hierarchical document structure analysis. *ArXiv*, abs/2401.11874.

Ross Wilkinson. 1994. Effective retrieval of structured documents. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jiawen Xie, Pengyu Cheng, Xiao Liang, Yong Dai, and Nan Du. 2023. Chunk, align, select: A simple long-sequence processing method for transformers. *ArXiv*, abs/2308.13191.

Hangdi Xing, Feiyu Gao, Rujiao Long, Jiajun Bu, Qi Zheng, Liangcheng Li, Cong Yao, and Zhi Yu. 2023. Lore: Logical location regression network for table structure recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(3):2992–3000.

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. 2021. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. Layoutlm: Pre-training of text and layout for document image understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Xu Zhong, Elaheh Shafieibavani, and Antonio Jimeno-Yepes. 2019a. Image-based table recognition: data, model, and evaluation. *ArXiv*, abs/1911.10683.

Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. 2019b. Publaynet: Largest dataset ever for document layout analysis. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *ArXiv*, abs/1904.07850.

A Appendix

A.1 Statistics of Layout Elements

DocHieNet contains 1673 documents with 15610 pages and more than 187K layout elements. Tab. 8 summarizes the overall frequency and distribution of different types of layout elements in DocHieNet.

class	count	%	class	count	%
title	2686	1.43	sidebar	383	0.20
sub-title	1435	0.76	table-title	944	0.50
section-title	20452	10.9	table	2244	1.20
text	116172	61.8	table-caption	1013	0.54
formula	709	0.38	header	8837	4.71
TOC-title	262	0.14	footer	6614	3.52
TOC	2011	1.07	footnote	3429	1.83
figure-title	1495	0.80	endnote	3402	1.81
figure	4547	2.42	page-number	9269	4.94
figure-caption	1694	0.90			

Table 8: Overview of the class of layout elements in DocHieNet. Along with the numbers of each class label, we present the relative occurrence

A.2 Details of Data Splits

Below are the detailed statistics of the data splits (See Tab. 9). As described in Sec. 4.3, the documents in the test set are fully annotated, whereas in the training set, 835 documents are only partially annotated. Consequently, the average number of pages per document in the training set is less than that in the test set. By establishing such a scenario, DocHieNet encourages DHP models to consider addressing the document inputs with various lengths encountered in real-world scenarios.

Split	#Docs	#En	#Zh	#Pages	#A.P.
train	1512	990	522	13299	8.8
test	161	120	41	2311	14.4

Table 9: Data split counts of DocHieNet. #En and #Zh respectively denote the quantities of English and Chinese documents, while A.P. signifies the average number of pages per document.

A.3 Details of Evaluation

We employ both F1-score to measure the correctness of predicted relation triples (Rausch et al., 2023) and Tree-Edit-Distance based Similarity (TEDS) to assess the entire document tree structure (Zhong et al., 2019a; Hu et al., 2022). Specifically, suppose $R_{gt} = \{(E_{parent}, E_{child}, r_{gt})\}$ and $R_{pred} = \{(\hat{E}_{parent}, \hat{E}_{child}, \hat{r}_{pred})\}$, then the F1-score is computed from the precision p_{score} and recall r_{score} as following:

$$p_{score} = \frac{|R_{gt} \cap R_{pred}|}{|R_{pred}|}, r_{score} = \frac{|R_{gt} \cap R_{pred}|}{|R_{gt}|}$$

Regarding TEDS, for the document D, a tree-like representation T_D can be obtained according to the hierarchical relations R, similar to a table of contents. Subsequently, the TEDS associated with the predicted structure $\hat{T}_D$ is calculated as follows:

$$TEDS(T_D, \hat{T}_D) = 1 - \frac{EditDist(T_D, \hat{T}_D)}{max(|T_D|, |\hat{T}_D|)} \quad (6)$$

A.4 Details of Baselines

We assess a group of DHP models to investigate their performance across different datasets. DocParser (Rausch et al., 2021) uses heuristics to convert a list of elements into hierarchical relations. It takes into account multi-column layouts but ignores most meta-information such as text content of elements. DSPS (Ma et al., 2023) employs a multi-modal encoder and a GRU (Chung et al., 2014) decoder for hierarchical organization. The textual embeddings of layouts are extracted seperately. And DOC (Wang et al., 2024) employs unified relation predictions to perform document layout analysis and hierarchy parsing from text lines. DSG (Rausch et al., 2023) leverages a bidirectional LSTM for relation prediction of the layout elements, employing features extracted from FPN for image regions and the GLoVe (Pennington et al., 2014) word embeddings of their layout element type.

A.5 Model Performance on Document of Different Languages

We have examined the performance of DHFormer on documents in languages of both English and Chinese, as illustrated in the Tab. 10. DHFormer exhibits stable performance on documents across

Split	DocHieNet-en		DocHieNet-zh	
metric	F-1	TEDS	F-1	TEDS
DHFormer	78.13	58.02	76.92	56.53

Table 10: The model performance on subsets of English and Chinese documents

different languages, though its performance on Chinese documents is slightly inferior. This is resulted by the fact that the pre-training data for the text-layout encoder of DHFormer is predominantly composed of English documents. Nevertheless, the layout knowledge acquired during pre-training proves effective for documents in both languages.

A.6 Details of LLM Implementations

A.6.1 APIs and Pre-trained Models

We employ two baselines for the discussion on LLMs: `GPT-4-turbo-128K` and `Llama2` (Touvron et al., 2023). GPT-4 represents one of the current state-of-the-art LLMs and is accessible via the OpenAI API[3]. Llama2 is a prevalent open-source large model in academia. The specific pre-trained model weight we utilize, *Llama-2-7b-chat-hf*, is available on Huggingface[4]. It has the original context length of 4096, and we extend it to 32K with position interpolation for the long document inputs.

A.6.2 Prompt for LLMs

To evaluate LLM on DocHieNet of document hierarchy parsing task, we define the prompt template as shown in Tab. 11. For fine-tuning Llama2, the ICL demonstrations are removed.

A.6.3 Fine-tuning Process of Llama2

Here we provide a detailed description of the fine-tuning process of Llama2. To cater for ability of Llama2 gained from pre-training, the DocHieNet dataset is transformed into a prompt-based format as illustrated in Tab. 11. The input document is organized as a list of layout elements arranged in reading order; and thus, the task is transformed into predicting the parent node of each element. The answer is organized as a list of relation pairs (i:j) as in Tab. 11. During training, the input is spliced into sub-documents within 10K tokens, and during testing, the input is the whole document. We follow the training hyper-parameters as demonstrated in llama-recipes [5]. We employ LoRA (Hu et al., 2021) for parameter-efficient fine-tuning, where we set the rank as 8, alpha as 32, dropout as 0.05, and the target modules are the query and value projections in the attention mechanism. The fine-tuning is done on 2 NVIDIA A100 GPUs for 1 epoch. We parse relationship pairs from the output, and reconstruct the document hierarchy trees based on these pairs. Essentially, all outputs are automatically parsable except for a handful of cases for which we make modifications manually.

[3] https://platform.openai.com/

[4] https://huggingface.co/meta-llama/Llama-2-7b-chat

[5] https://github.com/meta-llama/llama-recipes

Prompt	Here is a list whose elements represent the content blocks of a document, and the indication of keys are as following: "text": A string representing the text in the content block. "page": An integer indicating the page number on which the content block appears. "id": An integer that uniquely identifies the content block. "box": the layout information of the content block. Documents are organized as a tree-like structure. Please find the parent element of each content block based on the text and layout of them. The format of your reply: [{$id_1 : parent\_id_1$},...,{$id_n : parent\_id_n$}] . And do not reply other content. Here are some demonstration: {Demonstrates} Here is the input document:{Input} — reply:	
Slots	Input	List of document layout entities from DocHieNet.
	Demonstrates	The selected demonstration with ground truth response.

Table 11: The prompt for evaluating LLMs on DocHieNet.