DocAtlas: Multilingual Document Understanding Across 80+ Languages

Anonymous ACL submission

Figure 1: **Overview of the DocAtlas framework.** (**Left**) Global script coverage across 80+ languages spanning 10 writing systems, illustrating the geographical and typological diversity of the corpus. (**Right**) Cross-lingual transfer performance after DPO training, showing consistent gains in both in-domain and out-of-domain accuracy across major OCR and vision-language models.

Abstract

Multilingual document understanding remains limited for low-resource languages due to scarce training data and model-based annotation pipelines that perpetuate existing biases. We introduce **DocAtlas**, a framework that constructs high-fidelity OCR datasets and benchmarks covering 82 languages and 9 evaluation tasks. Our dual pipelines, differential rendering of native DOCX documents and synthetic LaTeX-based generation for right-to-left scripts produce precise structural annotations in a unified *DocTag* format encoding layout, text, and component types, without learned models for core annotation. Evaluating 16 state-of-the-art models reveals persistent gaps in low-resource scripts. We show that Direct Preference Optimization (DPO) using rendering-derived ground truth as positive signal achieves stable multilingual adaptation, improving both in-domain (+1.9%) and out-of-domain (+1.8%) accuracy without measurable base-language degradation, where supervised fine-tuning degrades out-of-domain performance by up to 21%. Our best variant, DocAtlas-DeepSeek, improves +1.7% over the strongest baseline.

1 Introduction

Despite recent advances in vision-language models (VLMs), multilingual document understanding[1] remains challenging due to the scarcity of high-quality training data across diverse scripts and languages (Liu et al., 2024; Xu et al., 2022, 2021). While models achieve strong performance on English documents, extending this capability to low- and medium-resource languages is hindered by limited annotated data.

Current dataset construction approaches face critical limitations. Manual annotation (Jaume et al., 2019; Xu et al., 2022) provides high-quality labels but does not scale beyond a handful of languages. Synthetic generation (Yim et al., 2021; Journet et al., 2017) avoids human labor but requires extensive per-script configuration and struggles with complex structures such as nested tables or authentic formatting. Model-based pipelines (Pfitzmann et al., 2022; Cui et al., 2025; Li et al., 2022) use pre-

[1]We use *document understanding* to denote the full pipeline from page image to structured output encompassing text, layout, tables, formulas, charts, and reading order, extending beyond character-level OCR.

trained models to label documents, creating circular dependency where annotation quality is bounded by existing model performance. This perpetuates bias: models trained on English produce annotations that train the next generation of English-centric models. Rendering-based approaches (Weber et al., 2023; Zhang et al., 2025) sidestep learned detectors by extracting structure from document source files, but suffer from rendering drift due to lossy format conversion (e.g., LibreOffice), lack geometric alignment between text and bounding boxes, and provide no coverage for right-to-left scripts or structured chart annotation.

We introduce **DocAtlas**, a pipeline for generating multilingual OCR datasets with model-free structural annotations that extracts ground truth directly from document sources through *differential rendering*. Unlike single-pass colorization, pixel-wise subtraction of colorized and standard renderings disambiguates injected annotations from pre-existing document colors, yielding precise bounding boxes without learned detectors. Results are serialized in a unified DocTag format jointly encoding component type, geometry, and text content, enabling multi-task supervision across all languages. To address the underrepresentation of right-to-left (RTL) scripts, which suffer from PDF parser failures in bidirectional text, we implement a complementary synthetic pipeline that converts structured sources (EPUB, HTML, XML) into PDFs using LaTeX with explicit bidirectional control, generating 52K additional pages with the same annotation precision. Optional metadata enrichment (e.g., figure classification, page attributes) may use auxiliary models, but all core DocTag annotations are fully model-free.

Combining both pipelines yields a 360K-page corpus across 82 languages and a difficulty-stratified benchmark of 5,862 pages spanning 9 evaluation tasks: end-to-end page parsing, text recognition, table extraction, formula transcription, chart parsing, reading order, and three format-specific subtasks (chart→HTML, formula→LaTeX, table→HTML). We evaluate 16 state-of-the-art models, revealing that low-resource scripts see 40–60% accuracy drops, structured extraction saturates at 73% TEDS regardless of language, and chart parsing sharply separates OCR-specialized systems from general VLMs. We further compare adaptation strategies and find that DPO achieves stable cross-lingual transfer (+1.8% accuracy, <3% base-language degradation) where supervised methods exhibit catastrophic forgetting (up to 21%), with QKV-only LoRA providing optimal gain-preservation balance. Our contributions are:

Dataset	# Samples	# Lang.	Data Type	Annotation	# Tasks	Model-Free
FUNSD	199	1	Real	Manual	1	✓
XFUND	1.3K	7	Real	Manual	2	✓
PubLayNet	361K	1	Real	Rule-based	3	✓
DocLayNet	81K	4	Real	Manual	4	✓
DocBank	500K	1	Real	Model-based	3	✗
SynthTIGER	10M	1	Synthetic	Rule-based	1	✓
WordScape	9.5M	136	Real	Colorization	3	✓
DIT700K	700K	1	Real	Layout detector	3	✗
DocAtlas (ours)	**360K**	**82**	**Real+Synth(15%)**	**Diff. Rendering**	**9**	✓

Table 1: **Comparison of OCR training datasets.** DocAtlas achieves 82 languages (11.7× more than XFUND) and 9 tasks (3× more than competitors) through rendering-based annotation that eliminates model dependency.

- A differential rendering pipeline producing model-free annotations from 307K documents across 82 languages, addressing five limitations of prior rendering-based approaches (§3.1).
- A synthetic RTL pipeline generating 52K pages with precise annotations for underrepresented bidirectional scripts (§3.2).
- A difficulty-stratified multilingual benchmark of 5.8K pages across 82 languages and 9 tasks with unified metrics, enabling systematic cross-model comparison (§3.3).
- A systematic study showing DPO with rendering-derived ground truth outperforms supervised fine-tuning and closed-model distillation for cross-lingual transfer (§3.4).

2 Related Works

OCR Dataset Construction. Traditional datasets rely on manual annotation (FUNSD (Jaume et al., 2019), XFUND (Xu et al., 2022)), limiting scalability. Synthetic pipelines (SynthTIGER (Yim et al., 2021), DocCreator (Journet et al., 2017), Donut (Kim et al., 2022)) use forward generation with text at predefined positions; annotations are trivially correct but cannot capture real document complexity such as nested tables or authentic formatting. Large-scale model-based efforts (PubLayNet (Zhong et al., 2019), DocBank (Li et al., 2020), DIT700K (Zhang et al., 2025)) automate annotation using pretrained layout detectors, creating circular dependency where quality is bounded by existing model performance.

Rendering-based pipelines offer a middle path.

Benchmark	Year	Langs	E2E	RO	Text	Table	Formula	Chart
PubTabNet (Zhong et al., 2020)	2019	1	✗	✗	✗	✓	✗	✗
XFUND (Xu et al., 2022)	2022	7	✗	✗	✓	✗	✗	✗
Nougat (Blecher et al., 2024)	2023	1	✓	✗	✓	✓	✓	✗
READOC (Li et al., 2025b)	2025	27	✓	✓	✓	✓	✓	✗
OmniDocBench (Ouyang et al., 2025)	2025	2	✓	✓	✓	✓	✓	✗
DocAtlas (ours)	2025	82	✓	✓	✓	✓	✓	✓

Table 2: **Document parsing benchmarks.** DocAtlas offers 3× more languages and supports all major elements. E2E: End-to-End, RO: Reading Order.

WordScape (Weber et al., 2023) recovers layout from Common Crawl Word documents via colorization, but relies on LibreOffice conversion (introducing rendering drift from font substitution and text reflow), extracts text independently of bounding boxes without geometric alignment guarantees, treats charts as opaque figures, and provides no RTL script coverage. We build upon this paradigm but treat rendering as a *closed-form annotation function*: lossless MS Word rendering eliminates drift, pixel-wise differential subtraction disambiguates injected colors from pre-existing ones, and joint IoU-based text-geometry matching produces aligned DocTag annotations suitable for multi-task supervision. We detail these improvements in §3.1. Table 1 summarizes the landscape.

Multilingual Model Training. Extending OCR to new languages without degrading original performance remains challenging due to catastrophic forgetting (Luo et al., 2025). Parameter-efficient methods (e.g., LoRA (Hu et al., 2022)) have been used for OCR adaptation (Chung and Choi, 2025) with reduced memory. Component-level training (Dolphin (Feng et al., 2025), SmolDocling (Nassar et al., 2025)) focuses supervision on specific elements rather than full pages. DPO (Rafailov et al., 2023) shows promise in preserving base capabilities. We systematically compare these strategies and additionally disentangle the effect of training algorithm from dataset quality by comparing DPO with rendering-derived ground truth against GPT-4o distillation (§3.4).

Evaluation Benchmarks. Existing benchmarks vary in scope: PubTabNet (Zhong et al., 2020) focuses on tables, XFUND (Xu et al., 2022) covers 7 languages, while recent efforts (READOC (Li et al., 2025b), OmniDocBench (Ouyang et al., 2025), Docling-Eval (Auer et al., 2024)) expand task coverage but remain limited in language diversity (27, 2, and 4 languages). Evaluation fragmentation prevents direct comparison across systems. Table 2 shows no existing benchmark simultaneously covers diverse languages and comprehensive document elements for parsing evaluation.

3 Methods

To construct large-scale multilingual OCR supervision without model dependency, we developed complementary pipelines (Figures 2,4). The first processes native Word documents from Common Crawl, while the second synthesizes right-to-left documents to fill gaps in script coverage.

3.1 Pipeline A: Native Word Documents

Inspired by (Weber et al., 2023), we begin by parsing `.wat` metadata from Common Crawl to extract candidate `.doc`/`.docx` URLs. Canonicalization-based deduplication is applied within each snapshot, and a RocksDB (Meta Platforms, Inc., 2024) key–value store ensures cross-snapshot deduplication, filtering out 60-80% of redundant URLs. Once URLs are extracted, the corresponding files are downloaded along with provenance metadata. During this stage, unsafe documents, those containing macros, embedded objects, or encryption, are automatically discarded, as are oversized or zip-bomb-like archives. SHA-256 hashing is applied to ensure content-level deduplication and integrity, and any file that fails to open or exhibits corrupted structure is logged and removed.

After acquiring a clean set of documents, we recover structure directly from OpenXML markup. Components are identified from native tags and built-in styles (e.g., tables, figures) and further refined using heuristic cues such as font size and list patterns. To distinguish component types, we inject color codes via Word styling attributes, then render both colorized and uncolorized versions to PDF (Figure 2). Subtracting these two renderings pixel-wise yields precise per-category bounding boxes through OpenCV contour analysis (Bradski, 2000), producing high-quality, model-free annotations from rendering differences alone.

With structure recovered, we align textual content to its geometric layout. Text is extracted at the document level from OpenXML and at the page level using the Docling (Livathinos et al., 2025) rule-based PDF parser (analogous to PyMuPDF, not a neural model). Word-level boxes are then matched to component regions using intersection-over-union (IoU) containment. When components overlap, such as text boxes drawn over images, we resolve conflicts by prioritizing the component with

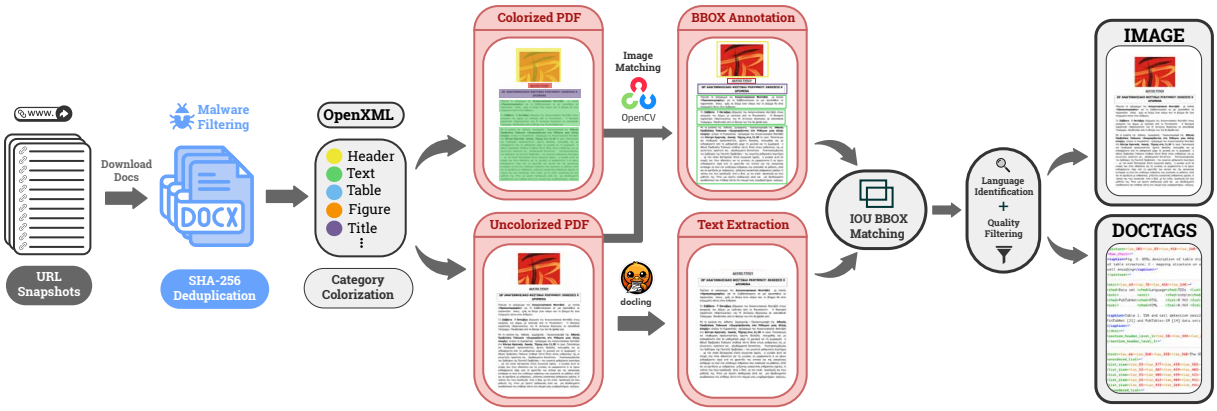

Figure 2: **End-to-end data pipelines.** We implement two pipelines: a high-fidelity pipeline for native DOCX documents and a synthetic RTL pipeline for underrepresented scripts. The native pipeline extracts, filters, colorizes, and annotates Word files, while the RTL pipeline converts structured inputs (EPUB, HTML, XML) into precisely annotated PDF documents using LaTeX synthesis.

higher style-based confidence, ensuring consistent structural mapping across complex layouts.

Quality Filtering and Perplexity Analysis. To maintain high multilingual quality, we apply a two-stage filtering process. First, we predict document language using fastText (Joulin et al., 2016) and compute perplexity via language-specific 5-gram Kneser–Ney models (Wenzek et al., 2020), thresholding at $\tau = 120$ to retain over 94% of high-quality data while filtering out 38% of low-confidence pages. Second, we compute an *annotation reliability score* based on the proportion of characters tagged via native XML signals rather than heuristics, excluding pages below 0.6 along with those exhibiting anomalous visual signals, resulting in roughly 15% removal following (Weber et al., 2023). Additional filtering details and per-language perplexity distributions are provided in Appendix 7.4. Real-world documents with complex backgrounds, nested tables, rotated text, or embedded advertisements are automatically detected and excluded to avoid propagating noisy supervision, preserving annotation precision at the cost of roughly 15% volume reduction.

Finally, we serialize all pages into the DocTag (Nassar et al., 2025) format, a unified XML-like schema encoding component type, geometry, and text content as shown in Figure 2. Unlike HTML, which omits layout geometry, or Markdown, which collapses hierarchy, DocTag preserves both structure and semantics, enabling multi-task supervision for layout detection, reading order, and content extraction. Each page becomes a flat tag sequence (e.g., `<text>`, `<section_header>`, `<table>`) with corresponding bounding boxes. To support flexible downstream use, we provide multiple output variants, including JSON, HTML, Markdown, and visual overlays.

Beyond basic annotations, we enrich each page with semantic metadata. Captions (`<figure_caption>`, `<table_caption>`) are identified through XML style cues and linguistic prefixes, then linked to nearest visual components via vertical adjacency. Figures are classified into categories (*natural image*, *logo*, *QR code*, *chart*, *graph*) via Docling (Livathinos et al., 2025), equations normalized to LaTeX, and page-level attributes (column count, watermark, background type) are inferred by Qwen3-VL (Yang et al., 2025). These two model-based steps provide *optional metadata enrichment only*, all core DocTag annotations are produced entirely through differential rendering and OpenXML parsing without learned models (Table 8).

Comparison with WordScape. Although our pipeline builds upon the Common Crawl extraction strategy of WordScape (Weber et al., 2023), the annotation methodology differs in three critical respects (Figure 3): (1) pixel-wise differential rendering disambiguates injected color codes from pre-existing document colors, which single-pass colorization cannot; (2) we render through MS Word rather than LibreOffice, eliminating stochastic drift from font substitution and text reflow; and (3) word-level IoU matching jointly encodes text, geometry, and component type, replacing fragmented JSON extraction with no alignment guarantees. Together, these enable multi-task supervision (layout + reading order + content extraction) rather than coarse layout detection alone.

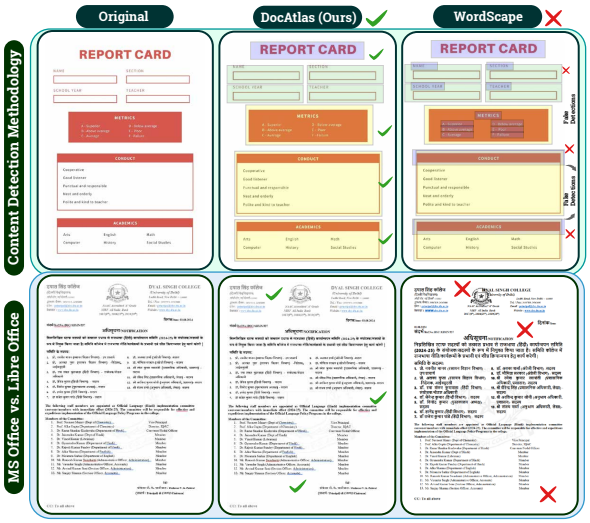

Figure 3: **Comparison with WordScape.** (Top) MS Word rendering preserves layout fidelity; LibreOffice introduces drift. (Bottom) Differential rendering eliminates false detections from pre-existing colors.

3.2 Pipeline B: Synthetic RTL Pipeline

While the native pipeline effectively covers left-to-right scripts, right-to-left (RTL) languages remain underrepresented due to parsing failures in existing PDF tools. To close this gap, we introduce a synthetic generation pipeline that produces near-perfectly annotated RTL documents through LaTeX-based rendering (Figure 4). Structured inputs (EPUB, HTML, XML) are parsed into a standardized Docling JSON schema, where each content element is tagged and assigned provisional bounding boxes. Document synthesis proceeds through 205 LuaTeX-based templates covering Arabic, Hebrew, Urdu, and Persian, governing typography, layout, and bidirectional text control:

$$\text{Docling} + \text{Template} \xrightarrow{\text{LaTeX}} \text{PDF} + \text{.pos.} \quad (1)$$

Custom LaTeX commands log positional metadata during three compilation passes (initial layout, position logging, final rendering), enabling exact bounding-box recovery for all elements. The resulting output pairs a rendered PDF with Docling (Livathinos et al., 2025) JSON containing element-level bounding boxes, text content, and structural labels. The pipeline generates 52K pages across 4 RTL languages with near-perfect annotation precision; implementation details including coordinate transformations, bidirectional markers, and chart synthesis are provided in Appendix 7.5.

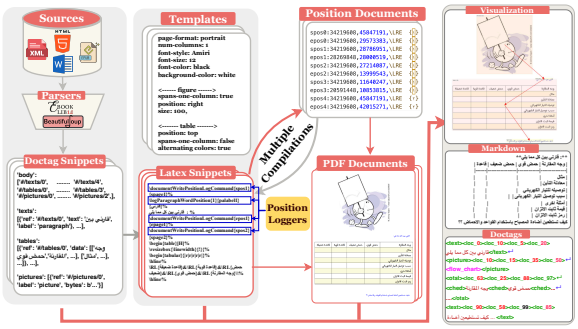

Figure 4: **Overview of the DocAtlas synthetic data generation pipeline.** Structured inputs (HTML, XML, DOCX, EPUB) are parsed into DocTag snippets and rendered via LaTeX templates with positional logging. Through multiple compilations, the system produces aligned PDF documents and precise element-level annotations (DocTag, Markdown, and visual overlays).

3.3 Benchmark

We assembled a multilingual benchmark balancing diversity, difficulty, and representativeness. Samples are drawn from the training corpus and targeted additions emphasizing rare structures (charts, formulas, multi-task layouts). Following (Ouyang et al., 2025), pages are embedded with ResNet-50 (He et al., 2016) features, clustered via FAISS (Douze et al., 2025), and stratified by difficulty into equal easy/medium/hard splits, yielding up to 100 pages per language across 82 languages (5,575 samples). We curate 144 challenging formula samples and generate multilingual chart data across 15 languages using a VLM-seeded pipeline with expert verification (κ=0.89; details in Appendix 7.7). Evaluation covers end-to-end page-to-Markdown/DocTag conversion, measured via text edit distance, TEDS (Zhong et al., 2020) for tables, formula transcription accuracy, and reading order fidelity. Additional subtasks, chart→HTML, formula→LaTeX, and table→HTML, extend evaluation to 9 tasks.

3.4 Multilingual Training Enrichment

We investigate three training strategies for extending OCR models to new languages while mitigating catastrophic forgetting: (i) full-page SFT on page→DocTag/Markdown pairs, (ii) component-level SFT on cropped elements (paragraphs, tables, charts, formulas), and (iii) DPO (Rafailov et al., 2023), which preserves base-language behavior by preferring rendering-derived ground truth over base model predictions. We further vary the subset of trainable parameters (QKV, MLP, or full model) to evaluate the gain-forgetting trade-off.

4 Analysis & Experiments

Dataset Statistics and Quality Control. We sourced 1.9M documents spanning 5.48M pages across 136 languages from Common Crawl under permissive licenses, with automated PII detection removing 5.15% of documents. Our native pipeline (Pipeline A) sustains 100k+ pages/day on a single CPU, while the synthetic RTL pipeline (Pipeline B) generates 195k pages at 183 pages/minute. Three document classes require targeted filtration, scanned PDFs (8.2%, excluded), rendering drift from missing fonts (<0.3%, mitigated via tolerance-aware contour matching), and malformed OpenXML (repaired via schema validation), ensuring 98.9% of retained documents maintain >95% annotation accuracy. After quality filtering and difficulty-aware sampling, the final corpus comprises 360k training pages across 82 languages, 31 structural element types, and 25+ content domains. Comprehensive details on collection, licensing, efficiency, and component distributions are provided in Appendices 7-7.2.

Model Selection and Evaluation Methodology. We evaluated 16 models spanning general VLMs (Hurst et al., 2024; Yang et al., 2025; Bai et al., 2025; Comanici et al., 2025; Wang et al., 2025) (multilingual baselines without layout training), expert document models (Nassar et al., 2025; IBM Granite Team, 2025; Xiaohongshu Hi Lab, 2025) (compact layout-grounded parsing), and OCR-specific systems (Wei et al., 2025; Mandal et al., 2025a,b; Feng et al., 2025; Cui et al., 2025; Li et al., 2025a) (cross-script supervision with structural output), enabling controlled analysis across architecture, scale, and training paradigms. We inference each model for Markdown outputs, then apply a three-stage pipeline (Ouyang et al., 2025): extraction (LaTeX/HTML tables, formulas, paragraphs with inline LaTeX→Unicode conversion), fuzzy Adjacency Search Match (Ouyang et al., 2025) using Normalized Edit Distance (direct matching for high-confidence pairs, iterative merging for partials), and metric computation across full-page parsing, individual tasks (text, table, formula, reading order), and condition-specific attributes (layout type, watermark, merged cells), ignoring headers/footers/captions. Detailed metrics are in Appendix 8.1. Additionally, training and data generation setups are in Appendix 8.2 and layout robustness in Appendix 8.3.

5 Results

5.1 Leaderboard Comparison

Our benchmark evaluation reveals critical performance patterns across multilingual document understanding. In Table 3, DocAtlas-Deepseek achieves state-of-the-art performance (83.37% overall), with DeepseekOCR following closely at 81.66% despite being a compact 3B model, demonstrating remarkable efficiency in balancing model size with accuracy. Notably, text recognition substantially outperforms structured content extraction across all systems: text edit distances average 0.068–0.095 for top models, while table TEDS scores plateau at 71–73%, highlighting that spatial reasoning over complex layouts remains a fundamental challenge. We identify 88,036 errors across 12 categories, with four dominant types: table spanning errors (15.7%), formatting (14.6%), character encoding (13.2%), and content omission (13.2%). These affect table structure, text styling, Unicode normalization, and list/hyphen handling.

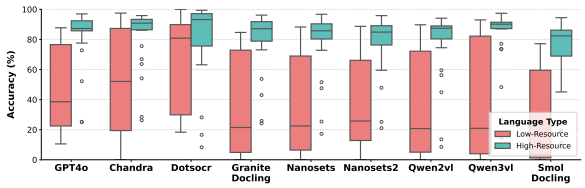

Figure 5: **Accuracy distribution across high- and low-resource languages.**

Figure 5 exposes a stark resource divide: high-resource languages maintain consistent 80-95% accuracy with narrow variance, while low-resource scripts exhibit 20-85% accuracy ranges with median performance often below 40%, underscoring how training data availability dictates multilingual robustness more than architectural sophistication. Cross-linguistic and domain-specific analysis re-

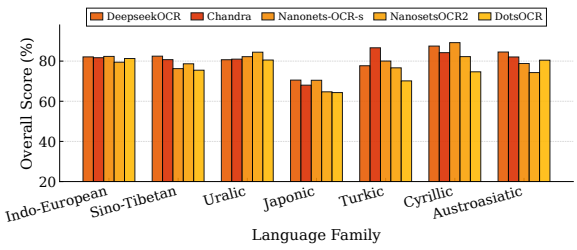

Figure 6: **OCR accuracy across language families.** Scores (brighter is better) show average performance across 14 models and 7 families. Top models (e.g., DeepseekOCR, Chandra) are consistent, while others degrade on low-resource scripts.

Method Type	Methods	Params	Text^{Edit}↓	Table^{TEDS}↑	Formula^{Edit}↓	Read Order^{Edit}↓	Overall^{Edit}↓
General VLMs	Gemini-2.0-Pro (Comanici et al., 2025)	-	0.090	68.50	0.356	0.050	79.75
	GPT4o (Hurst et al., 2024)	-	0.117	62.26	0.425	0.065	75.30
	Qwen3-VL (Yang et al., 2025)	3B	0.081	51.86	0.420	0.089	71.87
	Qwen2.5-VL (Bai et al., 2025)	2B	0.174	50.59	0.453	0.117	66.59
	InternVL3.5 (Wang et al., 2025)	2B	0.095	70.80	0.543	0.060	77.20
Expert VLMs	SmolDocling (Nassar et al., 2025)	256M	0.201	37.25	0.706	0.118	58.55
	Granite-Docling (IBM Granite Team, 2025)	258M	0.110	42.32	0.615	0.068	56.11
	DotsOCR (Xiaohongshu Hi Lab, 2025)	3B	0.068	65.40	0.321	**0.037**	79.29
	PaddleOCR-VL (Cui et al., 2025)	1B	0.078	73.90	0.241	0.052	80.10
	DeepseekOCR (Wei et al., 2025)	3B	0.082	71.54	0.242	0.053	81.66
	MonkeyOCR-pro (Li et al., 2025a)	1.2B	0.095	72.80	0.295	0.065	78.25
	Dolphin (Feng et al., 2025)	400M	0.160	58.30	0.465	0.066	71.17
	Nanonets-OCR-s (Mandal et al., 2025a)	4B	0.088	71.90	0.518	0.059	81.53
	Nanonets-OCR2 (Mandal et al., 2025b)	3B	0.088	66.24	0.471	0.060	78.70
	Chandra (To, 2025)	9B	0.071	69.79	0.262	0.042	81.33
	MinerU2.5 (Niu et al., 2025)	1.2B	0.267	**72.79**	0.273	0.096	73.07
	DocAtlas-Deepseek (Ours)	3B	**0.055**	72.24	**0.237**	0.049	**83.37**

Table 3: **Quantitative comparison across OCR systems on our multilingual benchmark.** We report character-level text recognition accuracy ($\text{TEXT}_{\text{EDIT}}$↓), table structure accuracy using TEDS (Zhong et al., 2020) ($\text{TABLE}_{\text{TEDS}}$↑), formula transcription accuracy ($\text{FORMULA}_{\text{EDIT}}$↓), and reading order fidelity ($\text{READ ORDER}_{\text{EDIT}}$↓). $\text{OVERALL}_{\text{EDIT}}$↓ represents the average of text and table scores (after converting text edit distance to accuracy). DocAtlas achieves the highest overall performance, outperforming prior expert and general-purpose VLMs baselines.

veals systematic biases in current OCR training paradigms. Language family performance (Figure 6) shows Indo-European and Cyrillic scripts achieving 80-87% accuracy, contrasting sharply with Japonic (26.9-70.5%) and Austroasiatic families where even top models struggle, suggesting that morphological complexity and logographic systems expose fundamental gaps in visual feature learning.

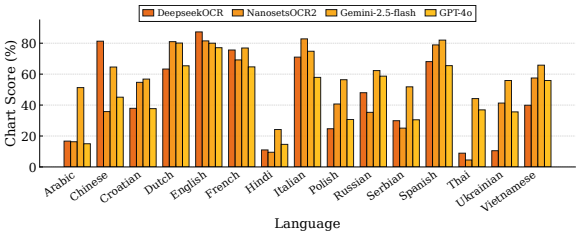

Figure 7: **Chart extraction accuracy across 15 languages.** Gemini-2.5-Flash achieves the highest average.

Model	Bar	Line	Pie
DeepseekOCR (Wei et al., 2025)	0.195	0.649	0.522
SmolDocling (Nassar et al., 2025)	0.127	0.038	0.337
NanosetsOCR2 (Mandal et al., 2025a)	0.397	0.603	0.446
Gemini-2.5-flash (Comanici et al., 2025)	**0.471**	**0.662**	**0.673**
GPT-4o (Hurst et al., 2024)	0.280	0.405	0.566

Table 4: **Chart type accuracy.** Mean OCR scores for each model across 3 chart types. Gemini-2.5-flash (Comanici et al., 2025) performs best overall.

Multilingual Chart Extraction Chart extraction reveals a critical divide between specialized OCR systems and general-purpose vision-language models. As shown in Figure 7, Gemini-2.5-Flash achieves the highest average performance (61.82%) with cross-lingual consistency, while expert OCR models exhibit severe language-specific degradation, DeepseekOCR scores 87% on English but collapses to 8-17% on Thai, Arabic, and Italian. Table 4 shows this gap persists across chart types, with multimodal models consistently outperforming document-specific architectures (e.g., SmolDocling's near-zero line chart accuracy of 0.038). These findings indicate that robust multilingual chart parsing requires visual reasoning beyond text extraction, motivating our inclusion of synthetic chart data across typologically diverse scripts.

Method	Text Edit Distance (↓)		Table TEDS (↑)	
	New Lang.	Base Lang.	New Lang.	Base Lang.
Full-SFT (All modules)	−0.038	+0.118	+13.6	−12.1
LoRA Variants:				
All layers	−0.025	+0.069	+8.9	−5.6
MLP only	−0.031	+0.042	+10.1	−3.9
MLP Gate&Down	−0.034	+0.028	+10.8	−2.7
All QKV	−0.027	+0.053	+9.7	−1.9
QKV only	−0.021	−0.011	+9.2	+1.3

Table 5: **Multilingual adaptation: gain vs. forgetting trade-off.** All changes measured relative to baseline (Text Edit: 0.082, Table TEDS: 71.5). LoRA (QKV only) achieves optimal balance.

5.2 Training Strategy Analysis

Layer-wise adaptation trade-offs. Table 5 shows that full SFT achieves the largest new-language gains (+13.6 TEDS) but severely degrades base-language performance (−12.1 TEDS). Among LoRA (Hu et al., 2022) variants, QKV-only training

achieves optimal balance (confirmed by (Zhu et al., 2025)): −0.021 edit distance on new languages while improving base performance (−0.011). We attribute this to a functional asymmetry: QKV governs *where* the model attends, while MLP layers shape output token distributions. Tuning MLP shifts distributions toward task-specific tokens, causing forgetting; QKV-only adaptation learns cross-lingual attention routing without biasing outputs. **DPO enables stable cross-lingual transfer.** Table 6 demonstrates that DPO (Rafailov et al., 2023) fundamentally breaks the adaptation-forgetting trade-off observed in supervised methods. While Full-Page SFT and Component-level training exhibit an inverse relationship between specialization and retention, DPO uniquely improves both metrics simultaneously. Component-level training achieves the highest in-domain gains but suffers catastrophic forgetting (up to -21.3%), suggesting that isolated document elements create brittle representations. This pattern holds across all architectures, indicating that using base model predictions as negative examples provides a fundamental mechanism for capability preservation.

Model	Baseline		Full-Page SFT		Component SFT		DPO	
	In	Out	In	Out	In	Out	In	Out
Qwen2.5-VL	66.6	61.9	70.5 (+3.9)	53.6 (-8.2)	71.2 (+4.6)	44.5 (-17.4)	**68.1** (+1.5)	**63.3** (+1.4)
Nanonets-OCR	81.5	75.7	86.3 (+4.8)	65.7 (-10.1)	87.1 (+5.6)	54.5 (-21.2)	**83.4** (+1.9)	**77.6** (+1.8)
DotsOCR	79.3	73.7	83.9 (+4.6)	63.9 (-9.8)	84.7 (+5.4)	53.0 (-20.7)	**81.1** (+1.8)	**75.4** (+1.8)
DeepseekOCR	81.7	75.9	86.4 (+4.8)	65.8 (-10.1)	87.3 (+5.6)	54.6 (-21.3)	**83.6** (+1.9)	**77.7** (+1.8)

Table 6: **Multilingual OCR adaptation strategies.** Performance on in-domain (In: trained languages) and out-of-domain (Out: unseen languages) test sets.

DPO Positive Signal	In-Domain	Out-Domain
Baseline (no DPO)	81.7	75.9
GPT-4o distillation	82.1 (+0.4)	75.2 (−0.7)
DocAtlas GT (ours)	**83.6** (+1.9)	**77.7** (+1.8)

Table 7: **Disentangling DPO from dataset quality.** DPO on DeepseekOCR using different positive signals. GPT-4o distillation hurts out-of-domain transfer due to biases on low-resource scripts; rendering-derived ground truth provides unbiased supervision.

Dataset quality drives DPO gains. To disentangle method from training signal, we compare DPO with rendering-derived ground truth against DPO with GPT-4o outputs as positives (Table 7). GPT-4o distillation yields marginal in-domain gains (+0.4) but degrades out-of-domain transfer (−0.7), as GPT-4o introduces systematic biases on low-resource scripts, hallucinated diacritics, RTL column misordering, that propagate through distillation. This validates that the annotation pipeline, not DPO alone, drives cross-lingual improvement.

Generalization to out-of-distribution documents. DPO-trained models also generalize beyond digital-native sources: on DocPTBench and OmniDocBench, both English-dominated, photographed/scanned benchmarks unseen during training, DocAtlas-DeepSeek reduces edit distance from 22.1%→20.7% and 0.137→0.122 over baseline DeepseekOCR, suggesting that cross-lingual attention routing learned via DPO transfers beyond the training domain.

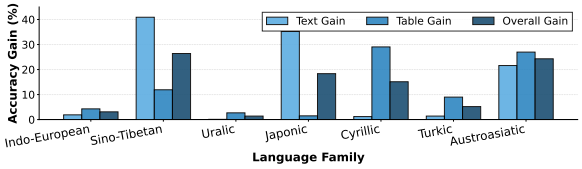

Figure 8: **DPO gains across language families.**

Language family gains reveal typological patterns. Figure 8 shows that DPO training benefits vary significantly across language families. Sino-Tibetan, Japonic, and Austroasiatic languages see large gains (e.g., 40% text for Sino-Tibetan), likely due to shared visual structures aiding transfer. Indo-European and Uralic languages show smaller gains (<5%), suggesting their scripts were already well-modeled. Cyrillic gains are skewed toward tables, indicating structured content transfers more easily than text.

We provide more results on the effect of document type in appendix 9.2), markdown evaluation analysis in appendix 9.3.

6 Conclusion

We presented **DocAtlas**, a pipeline constructing multilingual OCR datasets through differential rendering, producing 360K training pages and a 5.8K-page benchmark across 82 languages and 9 tasks without learned models for core annotation. Evaluation of 16 models reveals persistent low-resource gaps and a language-invariant table ceiling (73% TEDS), indicating spatial reasoning as the primary bottleneck. DPO with rendering-derived ground truth achieves stable cross-lingual transfer (+1.7% accuracy, <3% base degradation), outperforming stronger-model distillation, while QKV-only LoRA balances multilingual gains with capability preservation. Annotation quality, not model scale, bounds multilingual document understanding.

Limitations

Our differential rendering pipeline requires access to native document source files (DOCX or structured markup), and therefore cannot annotate scanned or photographed documents lacking digital text layers. This is by design, model-free annotation relies on source-level structure, but limits applicability to born-digital corpora. Combining DocAtlas supervision with OCR-from-scratch methods for scanned documents is a natural extension.

Use of Language Models

Large language models were used in a limited capacity to assist with minor editing and polishing of the manuscript, including improvements to clarity and grammar. All technical content, experimental design, results, and conclusions were produced, verified, and finalized by the authors.

References

Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, and 1 others. 2024. Docling technical report. *arXiv preprint arXiv:2408.09869*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lukas Blecher, Guillem Cucurull Preixens, Thomas Scialom, and Robert Stojnic. 2024. Nougat: Neural optical understanding for academic documents. In *International Conference on Learning Representations*, volume 2024, pages 37646–37663.

G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Yan Hon Michael Chung and Donghyeok Choi. 2025. Finetuning vision-language models as ocr systems for low-resource languages: A case study of manchu. *arXiv preprint arXiv:2507.06761*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

C. Cui and 1 others. 2025. Paddleocr-vl: Boosting multilingual document parsing via a 0.9b ultra-compact vision-language model. ERNIE Technical Report. Available at https://ernie.baidu.com/.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *IEEE Transactions on Big Data*.

Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, Jingqun Tang, Hao Liu, and Can Huang. 2025. Dolphin: Document image parsing via heterogeneous anchor prompting. In *Proceedings of the 65th Annual Meeting of the Association for Computational Linguistics (ACL)*. ArXiv:2505.14059.

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

Ahmed Heakl, Muhammad Abdullah Sohail, Mukul Ranjan, Rania Elbadry, Ghazi Shazan Ahmad, Mohamed El-Geish, Omar Maher, Zhiqiang Shen, Fahad Shahbaz Khan, and Salman Khan. 2025. Kitab-bench: A comprehensive multi-domain benchmark for arabic ocr and document understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22006–22024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

IBM Granite Team. 2025. Granite docling: A 258m-parameter multimodal vlm for document understanding. https://huggingface.co/ibm-granite/granite-docling-258M.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE. ArXiv:1905.13538.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Nicholas Journet, Muriel Visani, Boris Mansencal, Kieu Van-Cuong, and Antoine Billy. 2017. Doccreator: A new software for creating synthetic ground-truthed document images. *Journal of Imaging*, 3(4):62.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.

Chenxia Li, Ruoyu Guo, Jun Zhou, Mengtao An, Yuning Du, Lingfeng Zhu, Yi Liu, Xiaoguang Hu, and Dianhai Yu. 2022. Pp-structurev2: A stronger document analysis system. *arXiv preprint arXiv:2210.05391*.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.

Zhang Li, Yuliang Liu, Qiang Liu, Zhiyin Ma, Ziyang Zhang, Shuo Zhang, Zidun Guo, Jiarui Zhang, Xinyu Wang, and Xiang Bai. 2025a. Monkeyocr: Document parsing with a structure-recognition-relation triplet paradigm. *arXiv preprint arXiv:2506.05218*.

Zichao Li, Aizier Abulaiti, Yaojie Lu, Xuanang Chen, Jia Zheng, Hongyu Lin, Xianpei Han, Shanshan Jiang, Bin Dong, and Le Sun. 2025b. Readoc: A unified benchmark for realistic document structured extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21889–21905.

Yuliang Liu and 1 others. 2024. Ocrbench: On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*.

Nikos Livathinos, Christoph Auer, Maxim Lysak, Ahmed Nassar, Michele Dolfi, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Kasper Dinkla, Yusik Kim, and 1 others. 2025. Docling: An efficient open-source toolkit for ai-driven document conversion. In *AAAI Conference on Artificial Intelligence*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

Souvik Mandal, Ashish Talewar, Paras Ahuja, and Prathamesh Juvatkar. 2025a. Nanonets-ocr-s: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging.

Souvik Mandal, Ashish Talewar, Siddhant Thakuria, Paras Ahuja, and Prathamesh Juvatkar. 2025b. Nanonets-ocr2: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging.

Meta Platforms, Inc. 2024. RocksDB: A persistent key-value store for fast storage environments. https://github.com/facebook/rocksdb. Accessed: 2024.

Sebastian Nagel. 2023. Common crawl: Data collection and use cases for nlp. *HPLT & NLPL Winter School on Large-Scale Language Modeling and Neural Machine Translation with Web Data, February*, 6.

Ahmed Nassar, Matteo Omenetti, Maksym Lysak, Nikolaos Livathinos, Christoph Auer, Lucas Morin, Rafael Teixeira de Lima, Yusik Kim, A Said Gurbuz, Michele Dolfi, and 1 others. 2025. Smoldocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21972–21983.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, and 1 others. 2025. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. In *The 64th Annual Meeting of the Association for Computational Linguistics–Industry Track*.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, and 1 others. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout analysis. *arXiv preprint arXiv:2206.01062*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. Layoutparser: A unified toolkit for deep learning based document image analysis. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*, pages 131–146. Springer. ArXiv:2103.15348.

Datalab To. 2025. Chandra: Ocr model that handles complex tables, forms, handwriting with full layout. Open-source code, Apache 2.0 license.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Maurice Weber, Carlo Siebenschuh, Rory Butler, Anton Alexandrov, Valdemar Thanner, Georgios Tsolakis, Haris Jabbar, Ian Foster, Bo Li, Rick Stevens, and 1 others. 2023. Wordscape: a pipeline to extract

multilingual, visually rich documents with layout annotations from web crawl data. *Advances in Neural Information Processing Systems*, 36:26048–26068.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012.

Xiaohongshu Hi Lab. 2025. dots.ocr: Multilingual document layout parsing in a single vision-language model. https://github.com/rednote-hilab/dots.ocr.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2022. Xfund: A benchmark dataset for multilingual visually rich form understanding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3214–3224.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Moonbin Yim, Yoonsik Kim, Han-Cheol Cho, and Sungrae Park. 2021. Synthtiger: Synthetic text image generator towards better text recognition models. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III 16*, pages 109–124. Springer. ArXiv:2107.09313.

Zhiyang Zhang, Yaping Zhang, Yupu Liang, Lu Xiang, Yang Zhao, Yu Zhou, and Chengqing Zong. 2025. From chaotic ocr words to coherent document: A fine-to-coarse zoom-out network for complex-layout document image translation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10877–10890.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: Largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE. ArXiv:1908.07836.

Zhen Zhu, Yiming Gong, Yao Xiao, Yaoyao Liu, and Derek Hoiem. 2025. How to teach large multimodal models new skills. *arXiv preprint arXiv:2510.08564*.

7 Data Generation Details

7.1 Dataset Statistics and Composition

The raw DocAtlas corpus contains 1,011,501 documents spanning 5.48M pages across 136 languages, sourced from two complementary pipelines: **Pipeline A (Native DOCX)**, comprising 1,002,465 documents and 5.29M pages across 136 languages, and **Pipeline B (Synthetic RTL)**, comprising 9,036 documents and 195K pages across 4 RTL languages (Arabic, Hebrew, Persian, Urdu). The data exhibits a long-tailed distribution: high-resource languages (English, Russian, Spanish) account for approximately 60% of total pages, while medium- and low-resource scripts such as Hebrew, Thai, Burmese, and Khmer each contribute over 50,000 pages. After quality filtering and difficulty-aware sampling (§3.3), the final corpus comprises 360K training pages across 82 languages and a 5,862-page evaluation benchmark.

Figure 9 illustrates the long-tailed language distribution in detail. English, Russian, and Spanish dominate the high-frequency tier, while medium- and low-resource scripts such as Hebrew, Thai, Burmese, and Khmer each contribute over 50,000 pages, ensuring meaningful representation across typological families. Figure 10 shows that DocAtlas spans over 30 structural element types. Dominant tags include `text`, `table`, and `heading_1`, while less frequent yet critical tags such as `equation`, `form_tag`, and `bibliography` provide supervision for rare but important document elements.

7.2 Domain Diversity

To ensure broad generalization, the DocAtlas corpus spans over 25 primary categories and subcategories (Figure 11), including *Health*, *Law & Government*, *Finance*, and *Science*. Coverage is balanced across professional domains (e.g., *Legal*, *Investing*), academic domains (e.g., *Biology*, *Education*), and public-interest domains (e.g., *Weather*, *Religion & Belief*), ensuring the corpus reflects both formal and informal document types encountered in practice.

7.3 Data License and Privacy Compliance

Licensing. All documents were sourced from publicly accessible web content and academic repositories under permissive licenses (CC-BY 4.0, CC0, public domain). Following established practices in web-scale dataset construction (Nagel, 2023; Weber et al., 2023), our crawler respects `robots.txt` directives and identifies itself via a documented user-agent with project contact information. Domains requiring opt-in permissions or explicitly prohibiting scraping were excluded, and no content was extracted from password-protected, paywalled, or authenticated sources.

Component	Method	Model-Free?
Text & bounding boxes	Differential rendering	✓
Tables & structure	OpenXML tags	✓
Captions & reading order	XML cues + adjacency	✓
Figure classification	Docling classifier	✗ (optional)
Page attributes	Qwen3-VL	✗ (optional)

Table 8: **Model dependency breakdown.** Core structural annotations are fully model-free. Two optional enrichment steps use learned models but do not affect DocTag output.

PII Category	Method	Precision
Personal names	NER (spaCy)	95%
Emails & phone numbers	Regex	99%
Government IDs	Regex + rules	99%
Addresses & coordinates	NER + Regex	96%
Financial identifiers	Regex	99%

Table 9: **PII detection categories and precision.**

Privacy Protection. We applied an automatic PII detection pipeline using Microsoft Presidio, configured with spaCy 3.5 and custom regex patterns for academic and professional documents. Table 9 summarizes the detection categories and their precision. Documents containing three or more PII instances, or any government-issued identifier, were automatically excluded, resulting in 942,118 removals (5.15% of 18.3M initially collected documents). Manual spot-checking of 1,000 retained documents confirmed a false-negative rate of 0.1%.

7.4 Pipeline A: Native DOCX Generation Efficiency

URL Extraction and Deduplication. We parsed three Common Crawl snapshots (CC-2023-14, CC-2023-06, CC-2021-43), extracting 11.4M raw URLs pointing to `.docx` and `.doc` files. Deduplication operated at two levels: (1) within-snapshot URL canonicalization (lowercase, parameter sorting, trailing slash removal), which reduced each snapshot by 60–80%, and (2) cross-snapshot deduplication via a RocksDB key–value store with SHA-256 URL hashing, yielding 3.4M unique URLs.

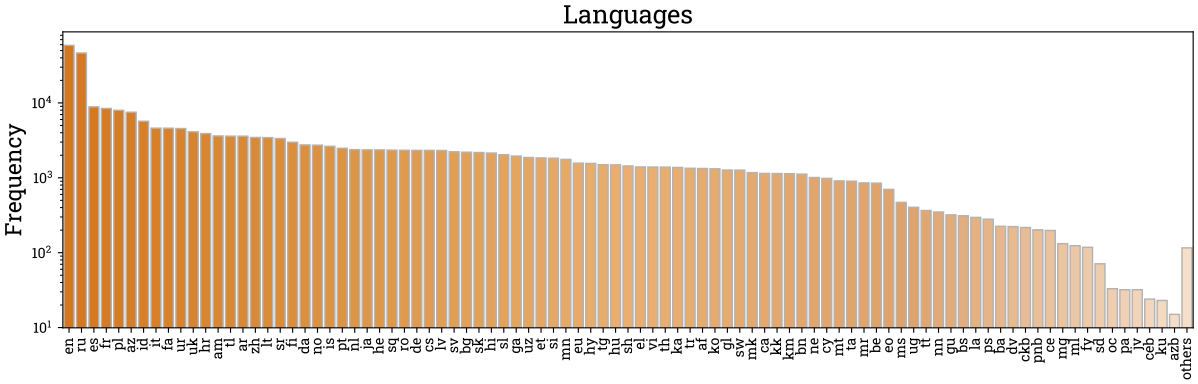

Figure 9: **Language frequency distribution in the DocAtlas corpus.** The dataset exhibits a long-tailed distribution across 80+ languages, with high-resource scripts (e.g., en, ru, es) dominating the head and low-resource languages (e.g., ps, ckb, ku, azb) forming a diverse tail.

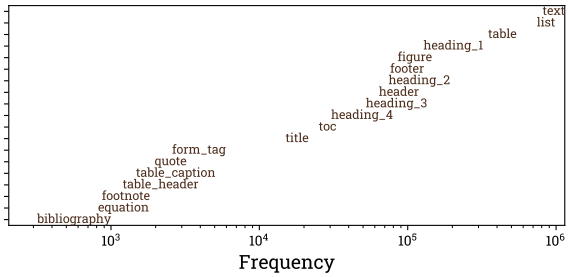

Figure 10: **Tag frequency distribution in DocAtlas.**

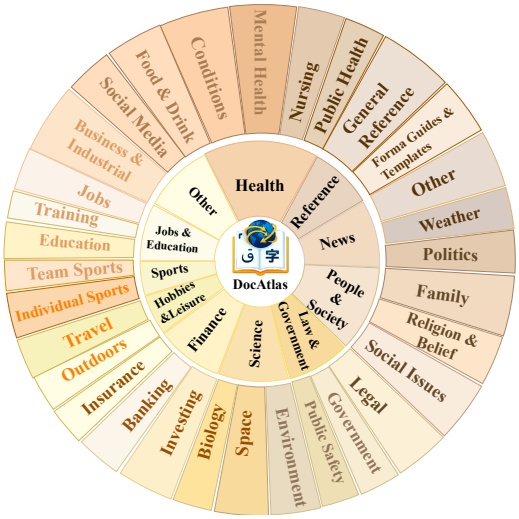

Figure 11: **Domain diversity across DocAtlas.** Each sector represents a major topic and its subdomains, showing balanced coverage across 25+ categories.

Download, Safety Filtering, and Annotation. Table 10 traces the collection funnel from raw URLs to the final annotated corpus. Of 5.8M HTTP requests issued (with exponential-backoff retries), 42.1% returned valid responses. Failures were dominated by dead links (38%), timeouts (12%), and redirects to non-document content (8%). Downloaded files were further filtered for Content-Type mismatches (14.9%) and malware indicators detected by OLETools (7.1%), producing a clean set of 1.9M DOCX files. During OpenXML parsing and differential rendering, an additional 19.9% of files were excluded due to corrupted ZIP archives (12.3%), extremely short documents under 200 characters (4.8%), and rendering engine crashes from unsupported features (2.8%). The final output is 1,002,465 structurally annotated documents spanning 5.48M pages across 136 languages.

Throughput. All processing was executed on a single Apple M2-Pro (10-core CPU, 16GB RAM) without multiprocessing, due to the MS Word singleton constraint. Table 11 reports per-stage timings. Sustained end-to-end throughput exceeds 100K annotated pages per day, completing 100K samples per snapshot in under 72 hours without GPU acceleration or distributed computing.

Quality Filtering and Perplexity Analysis. To maintain high multilingual quality, we apply a two-stage filtering process. First, we predict document language using fastText (Joulin et al., 2016) and compute perplexity via language-specific 5-gram Kneser–Ney models (Wenzek et al., 2020). As shown in Figure 12, thresholding at $\tau = 120$ retains over 94% of high-quality data while filtering out 38% of low-confidence pages. Second, we compute an *annotation reliability score*, defined as the proportion of characters successfully tagged via XML styles or table markers rather than heuristics. Pages with reliability below 0.6 or with anomalous visual signals (excessive blank space, corrupted

Stage	Count	Primary Removals
Raw URLs extracted	11.4M	—
After deduplication	3.4M	60–80% per snapshot
Valid HTTP responses	2.44M	Dead links, timeouts
After safety filtering	1.9M	Content-Type, malware
After parsing/rendering	1.0M	Corrupt ZIP, short docs

Table 10: **Pipeline A collection funnel.** Each stage shows the surviving document count and the primary removal reasons.

Operation	Time	Throughput
URL extraction	0.02s	4,320K pg/day
Doc. download	1.20s	72K pg/day
OpenXML parsing	0.15s	576K pg/day
Diff. rendering	0.45s	192K pg/day
Text extract. & align	0.12s	720K pg/day
Total (E2E)	**0.74s**	**117K pg/day**

Table 11: **Per-page processing time for Pipeline A.**

rendering) are excluded, resulting in roughly 15% removal of low-confidence pages following (Weber et al., 2023).

7.5 Pipeline B: Synthetic RTL Data Generation

Source Documents and Output. We collected structured documents (EPUB, HTML, XML) from open digital libraries and academic repositories for four RTL languages. Table 12 summarizes the input sources, template counts, and filtered output per language. The average document length is 3.5 pages.

Lang.	Source Docs	Templates	Output Pages	%
Arabic	4,636	64	123K	63.1
Hebrew	2,937	60	39K	20.0
Persian	1,378	36	26.3K	13.5
Urdu	85	45	6.7K	3.4
Total	9,036	205	195K	100

Table 12: **Pipeline B: RTL data summary by language.**

Template System. Each language-specific LaTeX template defines page format (portrait/landscape), column count (1–3), font family and size (9–14pt), colors, margins, header/footer styles, and bidirectional text handling primitives. Font selections reflect typographic conventions of each script: Amiri, Scheherazade, and Traditional Arabic for Arabic; David, Narkisim, and Frank Ruehl for Hebrew; Nazanin, Lotus, and Iranian Sans for Persian; and Nastaliq and Naskh for Urdu. Persian templates additionally support mixed LTR/RTL layouts for scientific content.

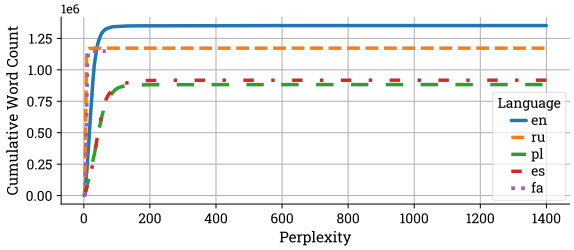

Figure 12: **Perplexity-based filtering across five languages.**

Rendering and Quality Control. The synthesis engine, built on LuaTeX with custom positional logging commands, operates in three compilation passes: (1) initial layout with placeholder positions, (2) position logging that writes exact element coordinates to `.pos` files, and (3) final rendering with validated positions. Sustained throughput is 183 pages/minute (10,980 pages/hour) on a single CPU core.

Quality filtering removes pages with positional inconsistencies exceeding 2pt coordinate drift between passes (15.2% of raw output), template misalignments such as overlapping elements or text overflow (8.9%), and font rendering failures including missing glyphs or incorrect shaping (2.1%). The filtered output of 195K pages demonstrates the scalability of the pipeline for producing bidirectional, script-accurate OCR supervision.

Input Parsing. Structured inputs are parsed into a standardized Docling JSON schema using BeautifulSoup[2] and ebooklib[3]. Each content element, headers, paragraphs, tables, and figures, is tagged and assigned provisional bounding boxes, which are later refined during LaTeX compilation.

Template Engine. The template engine governs typography (font family, size, color), layout (single or multi-column, margins, spacing), and positioning. We implement 205 LuaTeX-based templates covering Arabic, Hebrew, Urdu, and Persian, each supporting bidirectional text control. For mixed-language segments, Latin text is wrapped in explicit bidirectional control markers (`\textdir TLT`

[2]https://beautiful-soup-4.readthedocs.io/en/latest/
[3]https://docs.sourcefabric.org/projects/ebooklib/en/latest/

... `\textdir TRT`) to preserve visual ordering. The pipeline also handles structured tables with `colspan` and `rowspan` attributes, and programmatically generates charts (bar, pie, or stacked) from tabular data using Matplotlib[4] or Seaborn[5].

Coordinate Recovery. During compilation, custom LaTeX commands log positional metadata for each element, allowing exact bounding-box recovery. Coordinates in scaled points (sp) are successively transformed across coordinate systems: *LaTeX (bottom-left, sp) → PDF (bottom-left, pt) → Image (top-left, px)*. Three compilation passes, initial layout, position logging, and final rendering, guarantee positional stability. The resulting output pairs a rendered PDF with Docling (Livathinos et al., 2025) JSON containing element-level bounding boxes, text content, structural labels, and annotation confidence scores.

7.6 Dataset Statistics and Quality Control

We sourced 1.9M documents spanning 5.48M pages across 136 languages from Commoncrawl, with all data sourced from publicly accessible web content under permissive licenses. Automated PII detection removed 5.15% of documents containing sensitive content. Our native pipeline (Pipeline A) processes 100k+ pages/day, while the synthetic RTL pipeline (Pipeline B) generated 195k pages at 183 pages/minute. Additionally, our differential rendering pipeline achieves near-perfect annotation fidelity in most cases. However, three document classes require targeted filtration: (i) *Scanned PDFs* rendered as monolithic images without structural markup are detected via metadata analysis and excluded; (ii) *Rendering drift* from missing fonts (<0.3% incidence) causes subpixel geometry shifts, mitigated through tolerance-aware contour matching; (iii) *Malformed OpenXML* documents with corrupted namespaces are repaired via schema validation and XML normalization, with unrecoverable cases filtered post-validation. These quality controls ensure annotation precision across the retained corpus. After quality filtering and difficulty-aware sampling, these source documents yielded a curated corpus of 360k training pages across 82 languages. The resulting corpus exhibits diverse language and component distributions, with 31 structural element types spanning from high-frequency tags (`text`, `table`) to rare but critical elements (`equation`, `bibliography`). Lastly, to ensure broad generalization, the corpus spans 25+ domains including Health, Law & Government, Finance, and Science, with balanced coverage across professional, academic, and public interest categories.

7.7 Benchmark Construction Details

Difficulty-Stratified Sampling. Following the clustering protocol of (Ouyang et al., 2025), each page is embedded with ResNet-50 (He et al., 2016) features and clustered via FAISS (Douze et al., 2025) to promote visual variety. Within each cluster, a difficulty score combines per-component weights (e.g., table, formula, chart, density, font variety, image ratio), and sampling is drawn from a normal distribution to approximate equal easy, medium, and hard splits. This yields up to 100 pages per language across 82 languages (5,575 samples). To enrich the benchmark, we manually curate 201 additional PDFs containing challenging formulas, adding 144 samples.

Chart Data Generation. To enrich our benchmark with charts, we develop a pipeline inspired by (Heakl et al., 2025). Topics are generated using an expert VLM (Qwen3-8B-VL (Yang et al., 2025)), rendered to multiple chart types via Matplotlib or Plotly[6], and filtered using GPT-4o (Hurst et al., 2024) before expert verification by three domain specialists who cross-verified structural integrity, LaTeX formula alignment, and RTL reading order, achieving inter-annotator agreement of 94.2% (κ=0.89).

Page-Level Attributes. We further annotate pages with fine-grained attributes such as column layout, watermark presence, and background color, predicted via Qwen3-8B-VL and confirmed by human inspection. These attributes enable controlled evaluations under specific visual or layout conditions.

7.8 Annotation Failure Analysis

Although our differential rendering pipeline achieves near-perfect fidelity (98.9% of documents maintain >95% annotation accuracy), three document classes consistently degrade annotation quality and require targeted intervention: scanned or rasterized PDFs (8.2% of downloads, excluded), rendering drift from font substitution (<0.3%,

[4] https://matplotlib.org/
[5] https://seaborn.pydata.org/
[6] https://plotly.com/python/

largely mitigated), and malformed OpenXML markup (repaired where possible).

Scanned or Rasterized PDFs. Documents consisting of page-sized raster images without text layers yield only a single `<image>` tag and provide no structured supervision. We detect them via a two-stage process: metadata analysis (checking for full-page `/Type /Image` entries or absence of text operators) followed by image entropy thresholding (scanned content typically exceeds 6.5 vs. below 4.5 for rendered text). Approximately 8.2% of downloaded PDFs are flagged and excluded; these are logged separately for potential use in OCR-from-scratch benchmarks.

Rendering Drift. Missing or non-embedded fonts trigger substitution fallbacks that introduce subpixel horizontal shifts, line-wrap changes, and vertical spacing adjustments between the colorized and uncolorized renderings. Although rare ($<0.3\%$ overall), these cases are concentrated in mathematical papers and financial reports (10–15% of those categories). Our mitigation pipeline applies layout fingerprinting to detect structural mismatches between renderings, tolerance-aware contour matching with relaxed IoU and pixel-offset thresholds, and subpixel registration via phase correlation before differencing. Pages with residual misalignment exceeding 5% of components are conservatively excluded. This recovers 92% of affected documents; the remaining 8% (0.024% of the total corpus) are filtered.

Malformed OpenXML. Documents with unclosed or misordered tags, duplicated element IDs, or corrupted namespace declarations often pass basic ZIP integrity checks but fail during deterministic element-to-geometry mapping, producing symptoms such as misclassified tags, silent annotation loss, or incorrect nesting. We intercept these via ECMA-376 strict schema validation, apply automatic XML repair (tag balancing, ID deduplication, namespace normalization), and verify consistency by comparing expected versus recovered element counts after rendering. Documents whose annotation reliability score, the proportion of characters tagged via native XML signals rather than heuristics, falls below 0.6 are excluded. This process successfully repairs 73% of malformed documents; the remaining 27% (0.8% of the total corpus) are filtered. The repair step adds 0.08s average overhead per document.

Quality Validation. To verify annotation accuracy across all filtering stages, we manually spot-checked 500 random samples per language (70,000 total pages across 140 languages). Human annotators marked bounding boxes and labels as correct or incorrect, computing per-page accuracy as:

$$\text{Accuracy} = \frac{\text{Correct annotations}}{\text{Total annotations}} \times 100\%$$

Results show that 98.9% of retained documents achieve $>95\%$ annotation accuracy, with a mean of 97.8% ($\sigma = 2.1\%$) across the corpus. The 1.1% of documents falling below 95% are retained with warning flags.

8 Experiments

8.1 Metric Definitions

We evaluate model outputs using five complementary metrics spanning text, table, formula, chart, and full-page performance.

Normalized Edit Distance (NED). The NED metric measures the similarity between a predicted string p and a ground-truth string g:

$$\text{NED}(p, g) = 1 - \frac{D_{\text{Lev}}(p, g)}{\max(|p|, |g|)} \tag{2}$$

where $D_{\text{Lev}}(p, g)$ is the Levenshtein edit distance. NED ranges from 0 (completely dissimilar) to 1 (identical). We report $1 - \text{NED}$ as *TextEdit* in Table 3, so lower is better.

Tree-Edit-Distance-based Similarity (TEDS). For table evaluation, predicted and ground-truth outputs are converted to HTML and parsed as DOM trees:

$$\text{TEDS}(T_p, T_g) = 1 - \frac{D_{\text{tree}}(T_p, T_g)}{\max(|T_p|, |T_g|)} \tag{3}$$

where D_{tree} denotes the tree edit distance between DOM trees T_p and T_g. TEDS penalizes both structural errors (missing rows, merged cells) and textual discrepancies within cells.

Character Detection Matching (CDM). For formula evaluation, CDM measures spatial alignment between detected and ground-truth characters:

$$\text{Precision} = \frac{|\mathcal{M}|}{|\mathcal{P}|}, \quad \text{Recall} = \frac{|\mathcal{M}|}{|\mathcal{G}|} \tag{4}$$

$$\text{CDM} = 100 \times \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

where $\mathcal{P}$ and $\mathcal{G}$ are sets of predicted and ground-truth characters, and $\mathcal{M}$ denotes matched character pairs under a spatial or syntactic tolerance.

Chart Score. Since charts are first converted into structured HTML tables, we evaluate chart extraction accuracy using the same TEDS metric applied to the resulting table representations:

$$\text{ChartScore}(c_p, c_g) = \text{TEDS}\big(\texttt{html}(c_p),\ \texttt{html}(c_g)\big) \tag{6}$$

where $\texttt{html}(\cdot)$ denotes the chart-to-HTML-table conversion. This formulation captures both the structural layout (rows, columns, headers) and the textual content of the underlying data.

8.2 Setup

All training experiments use 4×A100 GPUs (40GB each), effective batch size 32, 3 epochs with cosine-scheduled learning rate of 2×10^{-5}, AdamW optimizer, LoRA rank 16 (α=32), and DPO β=0.1. Pipeline A throughput benchmarks were measured on a single Apple M2-Pro (10-core CPU, 16GB RAM), sustaining 100K+ annotated pages/day without GPU acceleration. To disentangle the contribution of our annotation pipeline from the DPO algorithm itself, we additionally compare against a distillation baseline where GPT-4o outputs serve as positive examples (§5).

Figure 13: **Qualitative comparison of layout parser failure cases.** Common error types (extra, overlapping, missing detections; wrong categories) are shown for Layout Parser (left) and our DocAtlas system (right). DocAtlas consistently produces more accurate and cleaner segmentation.

8.3 Layout Robustness

Figure 13 provides a qualitative comparison between traditional layout parsers (Livathinos et al., 2025; Huang et al., 2022; Shen et al., 2021) and the proposed DocAtlas annotation pipeline. Layout parsers often fail under visually complex conditions such as overlapping elements, nonstandard styles, or RTL scripts, resulting in fragmented bounding boxes and incorrect component classification. In contrast, DocAtlas maintains structural integrity even in pages with dense tables, multilingual content, or rotated elements, owing to its differential rendering and OpenXML-based annotation.

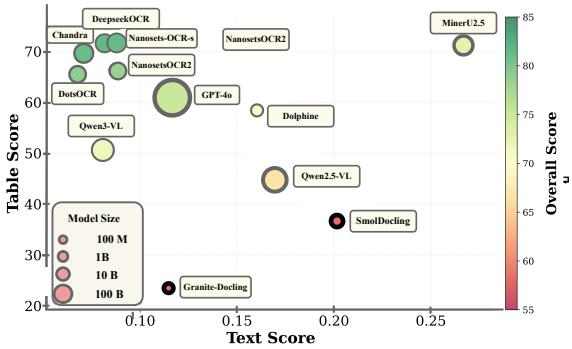

Figure 14: **Scores vs. model scale.** Each point represents a model; marker size encodes parameter count. Larger models do not uniformly dominate: several compact expert systems (≤3B) match or exceed general-purpose VLMs on both text and table scores.

8.4 Chart Extraction Evaluation Setup

To assess cross-lingual robustness in chart understanding, we evaluated OCR and vision-language models with native chart parsing capabilities. The evaluated systems include DeepseekOCR (Wei et al., 2025), SmolDocling (Nassar et al., 2025), NanosetsOCR2 (Mandal et al., 2025b), Gemini-2.5-Flash (Comanici et al., 2025), GPT-4o, and GPT-4o-mini (Hurst et al., 2024). We selected 15 representative languages: Arabic, Chinese, Croatian, Dutch, English, French, Hindi, Italian, Polish, Russian, Serbian, Spanish, Thai, Ukrainian, and Vietnamese. These languages cover diverse writing systems including Latin, Cyrillic, Arabic, Devanagari, and logographic scripts, representing both high-resource and regionally significant languages for real-world multilingual scenarios.

9 Results

9.1 Scores vs. Model Scale

Figure 14 visualizes the relationship between model scale and overall performance, showing that compact expert systems can rival much larger general-purpose VLMs.

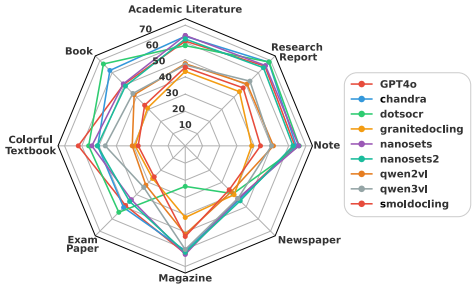

Figure 15: **Document type performance.**

9.2 Document Type Evaluation

(Figure 15) confirms training data composition directly shapes capabilities: models excel on academic literature and research reports (65-70% accuracy), domains heavily represented in datasets like DocBank (Li et al., 2020), yet collapse on newspapers and magazines (30-45%), where dense multi-column layouts and varied typography break learned priors. This performance asymmetry indicates that benchmark diversity, not just scale, determines real-world generalization.

9.3 Markdown Error Analysis

Systematic evaluation of 11 OCR models across 5,345 documents reveals 88,036 errors across 12 categories, with several dominant patterns.Spacing errors are most frequent (15.7%), primarily involving extraneous line breaks that fragment paragraphs. Formatting errors (14.6%) manifest as incorrect bold/italic tags and inconsistent dash characters (U+2212 vs U+2013). Character encoding errors (13.2%) involve Unicode normalization issues, particularly with ellipsis characters. Content omission (13.2%) affects hyphenated words and list separators. Structural challenges include table structure errors (8.3%) with incorrect `<thead>` insertion and list formatting errors (8.2%) with inappropriate `<br>` tags. Heading errors (7.1%) show systematic level misclassification, particularly H6 to H1/H2 conversions. Lower-frequency issues include content addition (7.1%), link errors (0.2%), image reference errors (0.7%), code block errors (0.6%), and math formula errors (0.1%).