Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Candidate-Routed LLM Control with Deterministic Reconstruction for Faithful Document Hierarchy Parsing

XiuJian Yan

1911342723@qq.com

Independent Researcher

Research Article

Keywords:

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11372414/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Candidate-Routed LLM Control with Deterministic Reconstruction for Faithful Document Hierarchy Parsing

XiuJian Yan[1*]

[1*]Independent Researcher, Shanghai, China.

Corresponding author(s). E-mail(s): 1911342723@qq.com;

Abstract

Purpose. The section hierarchy of born-digital documents underpins retrieval and question answering. Large language models (LLMs) can judge which lines are headings, but when they regenerate a document they may drop, paraphrase or invent content. We ask how much of an LLM's structural benefit survives when it may never transport content.

Methods. Constellation separates a control plane from a data plane. Deterministic providers turn DOCX/PDF files into ordered blocks with typed evidence. The LLM reads a sparse, candidate-centred view and returns heading anchors, which a document-wide constrained decoder fuses with deterministic proposals and with the model's silence on candidates it was shown. Closure assigns the original blocks to disjoint sections.

Results. On 454 Comp-HRDoc test documents with the PDF outline hidden, the LLM reader raises F1 from 0.776 to 0.881 and tree-edit-distance similarity (TEDS) from 0.644 to 0.791 over the deterministic core; without the silence channel the gain largely disappears. On a 100-document sample, its F1 is not significantly different from LLM rewriting, with 19× fewer output tokens, and it preserves 99.3% of the source words versus 96.7%. Docling detects headings more accurately (F1 0.925) and, re-levelled by its numbering, reaches a TEDS not significantly different from the full system's. In a post-hoc variant, Docling's detections and silences decoded as evidence raise Constellation to 0.937 F1 and 0.865 TEDS.

Conclusion. An LLM can act as an observer whose votes and silences are decoded with deterministic evidence, retaining much of its structural benefit with content ownership guaranteed.

Keywords: document structure analysis, logical layout analysis, table-of-contents extraction, large language models, deterministic reconstruction, Comp-HRDoc

1 Introduction

Converting DOCX and PDF files into structured representations is a core stage of digital libraries, enterprise knowledge management and retrieval-augmented generation (RAG)[1]. A useful intermediate representation must do two things at once: preserve the physical content exactly, and recover the logical hierarchy of sections and subsections that tells a reader, or a retriever, what each passage belongs to.

Existing parsers trade these goals against cost. Rule-based converters[2–4] are fast and deterministic, but they rely on explicit styles or stable visual conventions; real documents contain many "manual headings"—bold, enlarged, centred or numbered lines without a semantic style. Layout

models trained on page images[5–7] detect heading regions well but say little about their nesting. Supervised hierarchy models[8–10] recover trees, but they require in-domain training data. Prompting a general large language model (LLM) to rewrite a document as Markdown or JSON[11;12] brings semantic judgement without training, but it also makes every character a generated token: omissions, paraphrases and output truncation become structural risks, and output cost grows with document length.

Constellation is built on one rule: *the language model is never the transport channel for document content*. The original blocks form a data plane; the LLM emits only compact control observations (heading anchors) over a sparse view; a deterministic program assigns every original block to exactly one section. The question this paper answers empirically is how much of the structural value of an LLM survives this restriction. We compare this restricted reader with the deterministic evidence alone and with an LLM that rewrites the whole document, and we measure what each costs.

Contributions

1. A control/data-plane architecture for DOCX/PDF hierarchy parsing in which content ownership is exhaustive and non-overlapping for any anchor set (Proposition 1), so that heading errors can never delete or duplicate text.
2. An evidence-first candidate model and a document-wide constrained decoder that fuses typed deterministic evidence with physically aligned LLM votes and, symmetrically, with the LLM's *silence* on candidates it was shown (Sect. 3.4).
3. An outline-controlled evaluation on Comp-HRDoc[9]. It uses a frozen development/test split and external baselines (embedded bookmarks, Docling with and without numbering-derived levels, and generative LLM rewriting), with paired bootstrap intervals and an ablation of the silence channel. Exported-text preservation and model cost are measured identically for Constellation and the generative baseline. All deviations from the frozen protocol are disclosed (Sects. 4–5).
4. Evidence that the observer view extends beyond the LLM: a layout detector's detections and silences, decoded as typed evidence, yield a system more accurate than either source alone; this variant was added after the first test results (Sect. 5.3).

2 Related Work

Logical structure and table-of-contents extraction

Recovering a document's section hierarchy has been studied as book-structure extraction from digitised books[13], as title detection and table-of-contents (TOC) reconstruction in financial prospectuses[14], and, more recently, as hierarchical reconstruction of scientific papers. HRDoc[8] annotates 2,500 papers line by line with semantic classes and parent relations; Detect-Order-Construct[9] decomposes the task into page-object detection, reading-order prediction and TOC construction and introduces the Comp-HRDoc benchmark that we use; DocHieNet[10] extends hierarchy parsing to diverse multi-page documents. These systems are supervised on in-domain annotations and operate on page images and text lines. Constellation is training-free and targets born-digital files, where text extraction is reliable but heading semantics are not.

Document conversion toolkits

For scholarly PDFs, GROBID[15] shows how far engineered cascades can go. Docling[7] and MinerU[16] combine layout detectors trained on DocLayNet-style data[6] with reading-order and export rules; MarkItDown[3], Tika[2] and Mammoth[4] convert office formats deterministically. Such toolkits typically label heading regions but assign little or no depth; we use Docling as the external baseline.

Layout-aware and generative models

Layout-aware pretrained models[17–20] improved visually rich document understanding, and image-to-markup models such as Nougat[21] decode pages directly into Markdown. Generative conversion handles structure that rules cannot see, but every output character is generated, so content preservation is a statistical property. Constellation instead asks the model for compact observations and decodes them jointly with deterministic evidence under hierarchy constraints; the model never emits body text.

3 Method

Constellation has four stages (Fig. 1). Everything except the LLM call in Stage 3 is deterministic.

Notation

$B = (b_0, \ldots, b_{N-1})$ is the ordered block stream, $T \subseteq B$ the non-empty text blocks and β the body-font estimate. C_{full} is the uncapped candidate set and $C_{\text{route}} \subseteq C_{\text{full}}$ the LLM-facing table with cap κ. R is the set of blocks the LLM was shown as candidates, S the ordered decoding sites and $A = \{a_1 < \cdots < a_K\}$ the decoded heading anchors.

3.1 Stage 1: Block Model and Providers

A document is an ordered block sequence; each block carries a stable identifier, a type (text, table, image), its payload and provider metadata (page, bounding box, font size, weight, alignment, list numbering, style). DOCX blocks keep run-level formatting atoms; Markdown delimiters in source text are escaped before trusted formatting markers are added. The PDF provider is a deterministic text-and-layout normaliser: it estimates the body font by character mass rather than span count, joins small-caps fragments geometrically, assigns columns and labels marginal roles (headers, footers, footnotes); repeated running heads and footers, watermarks and printed tables of contents are set aside in a recorded audit stream rather than deleted, while footnotes and side notes stay in the content stream. It merges consecutive same-column, same-size lines into paragraphs unless a line carries heading evidence. Bold weight is read from font flags and from font names, since TeX fonts such as `CMBX` or `NimbusRomNo9L-Medi` carry weight only in their name. A PDF has no paragraph styles: the provider's internal font-size flag is therefore never reported as an explicit author style. The PDF outline (bookmarks), when present, is aligned monotonically to blocks by page and title; Sect. 5.2 evaluates the system with and without it.

3.2 Stage 2: Evidence-First Heading Candidates

Each text block receives typed evidence atoms with a polarity, a heuristic likelihood ratio λ_e, a reliability ρ_e and a correlation group; atoms that are surfaces of one cause (a heading style and its outline level) share a group, of which only the strongest is kept. With prior $\pi = 0.05$,

$$p_H(b) = \sigma\Big(\operatorname{logit}\pi + \sum_{e \in E^*(b)} \rho_e \log \lambda_e\Big). \qquad (1)$$

Here $E^*(b)$ is the set of evidence atoms of b after keeping the strongest atom of each correlation group. Reliabilities lie between 0.75 and 0.99 and are listed with the released code. Visible and list numbering, explicit heading styles and outline evidence are called *structural*. These are versioned heuristic scores, not calibrated posteriors. Table 1 lists the evidence families. Negative evidence is typed so that it can be audited: captions (including stand-alone labels such as "TABLE I"), list prose, table regions, printed table-of-contents lines (including dot leaders), marginal roles, document metadata (e-mail, date and arXiv stamps), entries of the bibliography zone, and *non-lexical* fragments—page numbers, equation numbers, numeric table rows and displayed formulas that contain no word. A section number must be followed by a word, so that a numeric table row such as "2.44 2.75" is not read as section 2.44. *Styled prose* is a block that wears a heading style but reads as a sentence; a *body signature* is a short styled line that has the body's size and weight. When an external layout detector is available, its heading regions enter as one more family: a block it reports as a heading receives positive evidence, and a text block on a page it read but did not report receives negative evidence, its silence. Neither is structural, so the reader's silence can still overrule a detection (Sect. 5.3).

Region diagnostics estimate the risk that local typography misses headings (no reliable cues) or is contaminated (cues everywhere) and classify each record as `strict`, `escape` or `audit_only`. The uncapped set C_{full} keeps every record for decoding and audit; only the LLM-facing projection is capped: records pass a probability floor (0.40 with strong structural evidence, but only 0.25 with weak evidence alone, so that regions without reliable typography still send candidates to the reader), and $\kappa = \max(30, \lfloor 0.15|T| \rfloor)$ is allocated with regional floors and risk-dependent extras.

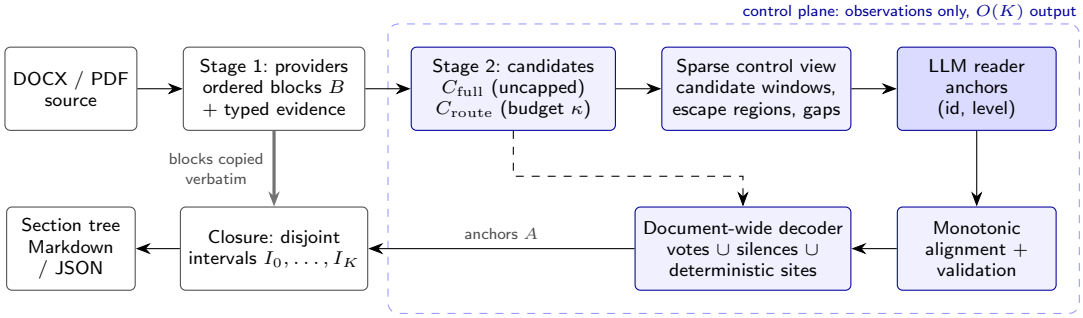

Fig. 1 Constellation. Content travels only along the data plane (black): providers emit ordered blocks, and closure copies them into section intervals. The control plane (blue) decides only where sections start and at which depth. The LLM sees a sparse, candidate-centred view and returns anchors; the decoder combines its votes, its silences on candidates it was shown, and the uncapped deterministic proposals (dashed).

Table 1 Evidence families (heuristic likelihood ratios λ_e of Eq. (1)).

Evidence family	Polarity	λ_e / condition
Explicit style (DOCX) / outline level / outline destination	+	120 / 90 / 150
Visible / list numbering (followed by a word; ≤ 140 characters)	+	30 / 18
Semantic title (Abstract, References, ...) / stand-alone line in a featureless region	+	30 / 30
Bold / centred / run-in cue	+	4 / 3 / 10
Font ratio $\text{fs}(b)/\beta$	±	4 at ≥ 1.10; 25 at ≥ 1.25; 35 at ≥ 1.50; 0.65 at ≤ 0.75
Caption / non-lexical / bibliography entry / printed TOC / margin	−	0.01 / 0.02 / 0.03 / 0.08 / 0.05
List prose / table region / styled prose / body signature / document metadata	−	0.12 / 0.18 / 0.05 / 0.05 / 10^{-4}
Layout detector: reported heading / silence on a read page (optional, Sect. 5.3)	±	30 / 0.10

3.3 Stage 3: Sparse Control View, Routing and Alignment

The control view expands the first eight blocks, every routed candidate with one neighbour on each side, and every complete `escape`-risk region; omitted spans become explicit `<Gap>` records with block-ID ranges, and preserved blocks carry relative meta-tags (bold, size above 1.05β, centring). The candidate table is appended as hypotheses with probabilities, level hints and region bands. The complete request—prompt, schema, view and table—is split only when it exceeds the configured input-token budget. The model returns a flat list of anchors (block id, title, level, snippet) and no body text.

Model outputs are observations, not trusted positions. A monotonic dynamic program aligns them to physical blocks using local windows, exact-text hits and a guarded remote rescue (at most eight options with similarity at least $\max(0.60, s_{\text{local}} + 0.20)$; at most 24 options per observation). After alignment, a vote is rejected if both its text similarity and its alignment score are low, and a vote landing outside every candidate is admitted only in an `escape` region and only if the block is short, title-shaped and similar to the reported title.

3.4 Stage 4: Document-Wide Decoding and Closure

Decoding sites are the union of C_{full} and the blocks that received valid votes. Each site s has the candidate probability $p_H(s)$ of Eq. (1), a contamination risk $\chi_s \in [0,1]$ from the region diagnostics; and a set V_s of valid votes. The log-odds that s is a heading are

$$z_s = \operatorname{logit} u_s + \mathbf{1}[s \in R,\, V_s = \emptyset] \log \lambda_{\text{sil}} + \sum_{v \in V_s} 0.9\, g_v \big(\operatorname{logit} w_v - \operatorname{logit} 0.18 \big). \quad (2)$$

The prior support u_s is $p_H(s)$ when the site has votes. Without votes it is the deterministic promotion support $p_P(s) = p_H(s)\,(0.55 + 0.45\,\bar{\rho}_s)(1 - 0.75\,\chi_s)$, where $\bar{\rho}_s$ is the largest reliability among the site's structural positive atoms (0.35 if none). When the site carries no negative evidence, $p_P(s)$ is raised to at least $0.70(1 - 0.75\chi_s)$ for a font ratio ≥ 1.25, or $0.64(1 - 0.75\chi_s)$ for a ratio ≥ 1.10 with bold. Two further rules apply. In regions

with $\chi_s \geq 0.65$ and no structural evidence, u_s is scaled by $\max(0.1, 1 - \chi_s)$. An unvoted site whose size and weight match at least two vote-confirmed sites receives $u_s \geq 0.62$. In a vote, $w_v = c_v \max(\alpha_v, 0.35)(1 - 0.45\chi_s)$ combines the vote's confidence c_v and its alignment score α_v, and $g_v \in (0, 1]$ halves the gain of each further vote from an overlapping request shard. The vote term compares each vote with an assumed 18% false-vote rate of the reader. Eq. (2) is a heuristic scoring rule in log-odds form, not a derived posterior.

Silence as an observation

If the reader was shown a candidate and did not mark it, that is evidence against it. Under a simple reader model with recall r and false-vote rate f, the likelihood ratio of silence is $\lambda_{\text{sil}} = (1-r)/(1-f)$; with $r = 0.9$ and $f = 0.18$ we fix $\lambda_{\text{sil}} = 0.12$ a priori. The vote term of Eq. (2) predates this model and is not its exact counterpart, whose log-ratio would be $\log(r/f)$. Both terms act in odds space, however, so a silence subtracts a constant $\log(1/0.12) \approx 2.1$ from the log-odds. A site with two independent structural cues, such as a heading style together with visible numbering, keeps $z_s > 0$ after one silence; a site with a single cue, including an outline destination alone, does not. Candidates that were never shown to the reader (beyond the budget) receive no penalty, so the budget cannot decide existence by itself.

Levels and decoding

With $h_s = \sigma(z_s)$, the conditional level score $q_s(k)$, $k = 1, \ldots, 6$, multiplies the candidate's level likelihoods, a vote bonus ($3.0\, g_v w_v$ at the voted level, $0.35\, g_v w_v$ at adjacent levels) and a document style-class prior. With this score scaled to its maximum, $\bar{q}_s(k) = q_s(k) / \max_j q_s(j)$, the emissions are

$$\psi_s(\text{NONE}) = 1 - h_s, \qquad \psi_s(\text{L}k) = h_s\, \bar{q}_s(k). \quad (3)$$

The emissions are not normalised over labels, so the decoder maximises a score rather than a posterior: it maximises $\sum_s \log \psi_s(y_s)$ over labels $y_s \in \{\text{NONE}, \text{L1}, \ldots, \text{L6}\}$ subject to the hard rule that the first heading is L1 and each heading is at most one level deeper than the previous one. When at least two sites carry section numbers, the state additionally keeps the stack of open ancestors (beam width 32) and subtracts a soft penalty, weighted by 2.8, for numbering that contradicts the proposed parent or sibling; otherwise exact Viterbi over the last level is used. No heading is inserted or re-levelled after decoding.

Closure

With decoded anchors $A = \{a_1 < \cdots < a_K\}$ and $a_{K+1} := N$, closure forms the preamble $I_0 = [0, a_1)$ and sections $I_j = [a_j, a_{j+1})$, and copies each block's payload from the data plane.

Proposition 1 (Exhaustive block ownership)**.** *Let A contain distinct valid block indices in increasing order. For non-empty A the intervals $I_0, I_1, \ldots, I_K$ partition $[0, N)$; for $A = \emptyset$ a single pseudo-root owns $[0, N)$. Every source block therefore belongs to exactly one section, independently of whether the headings and levels are correct.*

Proof. Consecutive half-open intervals share endpoints but no elements, and their union runs from 0 to N. For $A = \emptyset$ the claim holds by definition ($[0, N)$ is empty if $N = 0$). □

The argument is elementary; its value lies in what it removes from the error budget. Heading mistakes can mis-nest content but cannot delete or duplicate it. A run-in heading is a character span at the start of a block: that block starts the section, and its text is split into a title span and a body span whose union is the block text. Preserving the source text additionally requires that the provider extracts it completely, that assembly copies each owned payload once and that rendering is faithful; Sect. 5.4 measures the result against the PDF text layer.

Cost

Evidence extraction and view construction are linear in N; alignment costs $O(\sum_w K_w M^2)$ for K_w observations per request and $M \leq 24$ options; Viterbi costs $O(|S|\,|Y|^2)$ with $|Y| = 7$ labels (beam search $O(|S| \cdot 32 \cdot |Y|)$). Model output is $O(K)$ anchors rather than $\Theta(N)$ regenerated tokens.

4 Experimental Setup

4.1 Benchmark and Protocol

Data

Comp-HRDoc[9] provides, for each of the 500 test documents of HRDoc-Hard[8], the complete table of contents as a list of section lines with page, bounding box and a relation to an earlier line (*contain*: child; *equality*: sibling; *connect*: continuation of a multi-line title). We merge continuation lines into their title and derive levels from the relations; the annotations are used unchanged otherwise. Comp-HRDoc distributes annotations only, so we fetched each PDF from arXiv by identifier. Because the annotation does not record the arXiv version, a document is admitted only if at least 80% of its annotated section lines occur on the annotated page (±1) of the fetched PDF. Of the 500 documents, 484 were admitted; 15 failed the version check and one has no sections.

Frozen split

The first 30 admitted identifiers in lexicographic order form a development set; it contains every document inspected while the evidence rules of Sect. 3.2 were revised. The code was then frozen (tag `ijdar-frozen` in the code repository) and evaluated on the remaining 454 documents (6,737 sections), which had not been inspected.

Deviations from the frozen protocol

We report every change made after the test runs started.

1. One PDF page with tens of thousands of vector paths exposed a quadratic step in figure-region clustering. It was replaced by an exact near-linear version (tag `ijdar-perf`), which also added the no-silence ablation as a configuration switch. The output is identical, as verified by a randomised equivalence test against the original and by unchanged predictions on development documents.
2. Runs that failed were re-run once. The causes were memory exhaustion during parallel runs and API time-outs. In total 113 document runs were re-run, 91 of them for the full system; because the reader samples, those 91 documents received a fresh model sample. After the re-run, no document failed in any system. A document that still failed would have been scored as an empty output, never dropped. The re-run documents are listed with the results.
3. After an internal review, the scoring conventions below were changed for *all* systems: (i) the title convention; (ii) page location and level normalisation for the generative baseline; (iii) a like-for-like preservation measure. All systems were re-scored from their stored predictions; no system was re-run for this.
4. Three analyses were added after the first test results were known: the Docling + numbering baseline, the layout-evidence variant of Sect. 5.3 and the strict-matching sensitivity analysis. The variant's two likelihood ratios were fixed from the development split before it was run once on the test split (tag `ijdar-layout`).

Matching and metrics

A predicted heading matches a reference section if their pages differ by at most one and their normalised titles (case-folded, alphanumerics only, with and without the leading section number) reach a similarity of at least 0.85 (Python `difflib` ratio) after both are truncated to the length of the shorter. The prefix rule lets a run-in heading reported together with its paragraph match its reference, but it also accepts a truncated or extended title; Sect. 5.1 therefore also reports a strict rule. Matching is one-to-one and greedy by score. We report the following:

- micro precision, recall and F1 over headings;
- level and parent accuracy over matched pairs, where a pair is parent-correct if both are top-level or the predicted parent is matched to the reference parent; both are conditional on a match, so they compare poorly across systems with different recall;
- a tree-edit-distance similarity[22], $\mathrm{TEDS} = 1 - \mathrm{TED}(\mathcal{T}_p, \mathcal{T}_g)/\max(|\mathcal{T}_p|, |\mathcal{T}_g|)$ for the predicted and reference trees, computed with APTED[23] and unit costs.

Our TEDS is a variant of the released Comp-HRDoc script. That script labels nodes by their exact class and text; ours labels a matched prediction with its reference node and every other prediction uniquely, so that text-extraction differences are not counted twice. TEDS is averaged

over documents. Differences are the named system minus the comparator and are tested with a paired bootstrap over documents (5,000 resamples, 95% percentile intervals). Run-to-run variation of the sampling reader is not part of these intervals; it is reported separately.

Scoring conventions

The same conventions apply to every system.

- *Title.* Comp-HRDoc's tables of contents never contain the paper title. A first-page prediction that reproduces the title annotated in the line-level annotation, and carries no section number, is therefore ignored rather than counted as a false positive.
- *Front and back matter.* The reference includes titles such as Abstract, References or Acknowledgments for some documents but not for others. As a sensitivity analysis we also report scores with these titles removed from predictions and reference alike.
- *Preservation.* Content preservation is the share of the words of the PyMuPDF text layer that occur in a system's exported Markdown, counted as a multiset; we also report the share that occurs in order. Both use the same tokenisation and sequence matcher for every system.

4.2 Systems

- **Embedded bookmarks**: the PDF outline read verbatim, with its own levels and pages; no parsing. Of the 454 test PDFs, 65 have no outline.
- **Docling**[7], version 2.138: all heading items, i.e. every `section_header` (Docling returned no `title` items), with Docling's level and page. Optical character recognition (OCR) and table-structure recognition are disabled so that, like Constellation, it reads the born-digital text layer. Docling assigns level 1 to every heading of a PDF. We therefore also evaluate **Docling + numbering**, a derived baseline that re-levels Docling's headings from their visible numbering: "3.2.1" becomes level 3, "II." level 1, "A." level 2, and unnumbered headings level 1.
- **Generative LLM**: the same LLM receives the page-marked PyMuPDF text of the whole PDF and rewrites it as Markdown with `#`-headings (temperature 0, up to 32,768 output tokens). Headings are read from its output. Both prompts, the reader's and this one, are part of the released code.
  - Pages: because the model dropped the page markers in some outputs, each heading's page is located in the source text, from the previous heading's page onwards, and the marker page is used only when the heading is not found.
  - Levels: the model usually renders the paper title as the only level-1 heading, so a single leading level-1 heading is dropped and levels are shifted so that the shallowest is level 1. Without this normalisation its level accuracy is 0.401.
  - Both adjustments favour the baseline, which because of its cost is run on a seeded random sample of 100 test documents.
- **Constellation, deterministic**: Stages 1, 2 and 4 without any model observation.
- **Constellation, full**: the production pipeline with the LLM reader.

Constellation is evaluated with the PDF outline hidden from the provider ("no outline", the main setting) and visible. The LLM is `deepseek-v4.1-flash`, accessed in October 2026 through an OpenAI-compatible gateway with reasoning disabled. Constellation's reader samples at temperature 0.5, its production setting; the generative baseline uses temperature 0, the usual setting for faithful rewriting. The two also see different inputs: the reader sees blocks with relative typography tags, whereas the generative baseline sees plain text. Constellation uses its default configuration, including $\lambda_{\text{sil}} = 0.12$, which was fixed before any measurement.

5 Results

5.1 Hierarchy Recovery on Comp-HRDoc

Table 2 reports the main comparison.

Contribution of the reader

With the outline hidden, the LLM reader raises F1 from 0.776 to 0.881 and TEDS from 0.644 to 0.791.

Table 2 Section hierarchy recovery on the Comp-HRDoc test split (454 documents, 6,737 sections). P/R/F1: micro over headings; Level and Parent: accuracy over matched headings; TEDS: mean tree similarity. $F1^{\dagger}$/$TEDS^{\dagger}$: sensitivity analysis with front- and back-matter titles removed from predictions and reference. Outline: whether Constellation sees the PDF outline (bookmarks). Docling + numbering and the last two rows, which add Docling's detections as evidence (Sect. 5.3), were added after the first test results.

System	Outline	P	R	F1	Level	Parent	TEDS	$F1^{\dagger}$	$TEDS^{\dagger}$
Embedded bookmarks	visible	0.904	0.670	0.770	0.897	0.961	0.650	0.818	0.734
Docling 2.138	–	0.916	0.934	0.925	0.552	0.552	0.668	0.923	0.647
Docling + numbering	–	0.916	0.934	0.925	0.934	0.887	0.805	0.923	0.806
Constellation, deterministic	hidden	0.733	0.825	0.776	0.947	0.902	0.644	0.762	0.637
Constellation, full	hidden	0.886	0.876	0.881	0.957	0.936	0.791	0.877	0.797
Constellation, full without silence	hidden	0.696	0.888	0.780	0.954	0.900	0.656	0.762	0.646
Constellation, deterministic	visible	0.757	0.889	0.818	0.919	0.912	0.717	0.809	0.716
Constellation, full	visible	0.891	0.910	0.900	0.949	0.945	0.828	0.900	0.841
Constellation, deterministic + layout	hidden	0.934	0.898	0.916	0.930	0.908	0.810	0.918	0.819
Constellation, full + layout	hidden	0.950	0.924	0.937	0.958	0.941	0.865	0.940	0.881

- The paired differences are +0.105 for F1 (95% interval [+0.083, +0.128]) and +0.146 for TEDS ([+0.127, +0.166]).
- Most of the gain comes from precision (0.733 to 0.886), with a smaller gain in recall (0.825 to 0.876). Deterministic evidence over-proposes, and the reader's silence removes the proposals it does not confirm.
- The silence channel accounts for most of this gain: the full system exceeds its own ablation without silence (votes still used) by +0.101 F1 [+0.082, +0.121] and +0.134 TEDS [+0.118, +0.150].
- With positive votes alone, F1 (0.780) is close to that of the deterministic core (0.776): recall is higher (0.888 vs 0.825) but precision lower (0.696 vs 0.733). The ablation is a separate run of the reader, so its votes are a different sample. The reader's value therefore lies mainly in what it declines to confirm.

Comparison with Docling

Docling detects headings more accurately than the full system. Its F1 is 0.925, and the full system's difference is −0.044 [−0.056, −0.032], with Docling higher in both precision and recall. Docling, however, places every heading of a PDF at level 1, so its level accuracy (0.552) is simply the share of matched sections that are top-level, and the full system's TEDS exceeds Docling's by +0.123 [+0.104, +0.142]. Once its headings are re-levelled from their numbering, Docling reaches 0.805 TEDS. The full system's TEDS differs from this by −0.014 [−0.031, +0.003], which is not a significant difference.

Embedded bookmarks

The bookmarks, produced by the authors or by a LaTeX package such as `hyperref`, reach 0.770 F1 and 0.650 TEDS. They are often incomplete (recall 0.670), and 65 of the 454 test PDFs have none.

Sensitivity to front and back matter

Removing front- and back-matter titles from predictions and reference changes F1 by at most 0.018 for every system except the bookmarks (Table 2, last two columns). The bookmarks rise from 0.770 to 0.818, because outlines usually omit these titles while the reference sometimes includes them. The order of the Constellation variants and Docling is unchanged, except that the ablation without silence and the deterministic core tie at 0.762.

Sensitivity to the matching rule

Under a strict rule—whole titles must reach the same similarity, and section numbers must agree when both titles carry one—F1 falls by 0.033–0.054 and TEDS by 0.026–0.045 for every system. The full system reaches 0.827 F1 and 0.748 TEDS, against 0.729 and 0.612 for the deterministic core. The order is unchanged except at the top: Docling (0.884) and the full system with layout evidence (0.883) tie on F1, while the latter keeps the highest TEDS (0.820 vs 0.776 for Docling + numbering).

Fig. 2 shows that the reader's gain is not carried by a few documents. The full system's curves lie above the deterministic ones over most of the range, and the layout-evidence variant lies above both. Because the reader samples at temperature 0.5, we ran the full system four times in total on

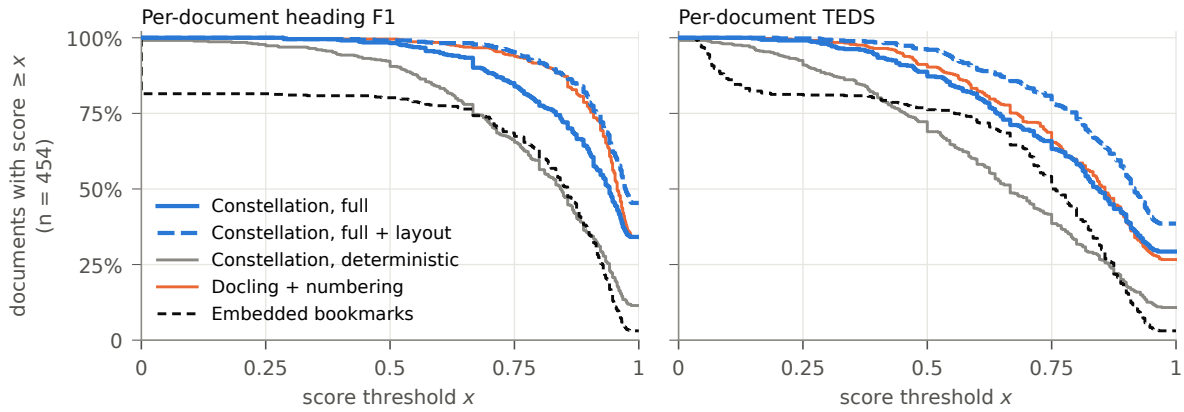

Fig. 2 Per-document score distributions on the test split; Constellation does not see the outline. Each curve gives the share of documents whose heading F1 (left) or TEDS (right) is at least x; higher is better. A document without bookmarks has F1 0 for the bookmark baseline and a small positive TEDS, because the root node always matches.

the 100 documents of the generative sample. F1 ranged from 0.886 to 0.890 and TEDS from 0.811 to 0.818.

Published supervised results

For context only: supervised models trained on the HRDoc-Hard training split report a document-averaged (macro) semantic tree-edit-distance similarity (STEDS) of 0.62–0.88 for TOC extraction on Comp-HRDoc[9]. Those numbers come from page images and in-domain training. They are computed with the released script, whose tree nodes carry exact class and text labels, rather than with our matched-label variant, so they are not directly comparable with the zero-shot, born-digital setting here.

5.2 Access to Embedded Outlines

PDF bookmarks are a strong but optional signal; when they are visible, they enter Stage 2 as outline-destination evidence.

- Exposing them changes the deterministic core by +0.041 F1 [+0.033, +0.050] and the full system by +0.019 [+0.013, +0.026].
- With bookmarks visible, the reader still adds +0.083 F1 [+0.062, +0.106] and +0.111 TEDS [+0.094, +0.129] over the deterministic core, so its contribution is not an artefact of hiding the outline.
- The bookmarks help the deterministic core more than the full system, plausibly because part of what they supply—which lines are headings—the reader already provides.

Comp-HRDoc's references are not derived from the bookmarks: they exist for every document, including the 65 without an outline. On benchmarks whose references are derived from the same outlines a parser can read, as in the long-document suite of Sect. 5.5, the two effects cannot be separated.

5.3 A Layout Detector as a Second Observer

Docling's advantage in detection suggested treating its output the way Constellation treats the reader, as observations rather than as a competing pipeline. Each of Docling's `section_header` items is matched to the block on the same page whose text starts with it. That block receives the positive layout evidence of Table 1, and every other text block on the page receives the negative evidence, a detector silence.

The two likelihood ratios were fixed on the development split before a single test run. The positive ratio of 30 was set equal to that of visible numbering rather than derived. The silence ratio follows the reader model of Sect. 3.4: with Docling's development recall $r \approx 0.93$ and a small false-detection rate f over all text blocks, $(1-r)/(1-f) \approx 0.07$–0.08, which we rounded up to the more conservative 0.10. Neither family

Table 3 Generative rewriting versus anchor routing (full system, outline hidden) on the same 100 test documents. Heading accuracy, tokens and time of Constellation come from its main run; its preservation from a repeat run, the only one that stored its export. Preservation: share of the words of the PDF text layer present in the exported Markdown, as a multiset (mean and worst document) and in order. Inserted: share of output words not in the source. Tokens and time are means per document; Constellation's time is given with its range over four runs.

	Generative	Constellation
Heading F1	0.902	0.888
TEDS	0.840	0.813
Words preserved (mean)	0.967	0.993
Words preserved (worst doc.)	0.692	0.863
Words preserved in order	0.960	0.953
Inserted words	1.1%	1.0%
Input tokens / doc.	12,030	11,893
Output tokens / doc.	10,868	574
Wall time / doc. (s)	44	20 (12–20)

is structural, so the reader's silence can still overrule a detection. On development documents the variant raised the full system's F1 from 0.849 to 0.868.

On the test split the variant gives the following results.

- Layout evidence alone lifts the deterministic core by +0.139 F1 [+0.118, +0.162] and +0.165 TEDS [+0.148, +0.183].
- With the reader, the system reaches 0.937 F1 and 0.865 TEDS. That is +0.056 [+0.045, +0.066] and +0.074 [+0.061, +0.086] above the full system without layout evidence (the text-only system).
- The reader still adds +0.021 F1 [+0.016, +0.027] and +0.055 TEDS [+0.043, +0.067] over the deterministic core with the same layout evidence.
- Against Docling with numbering-derived levels, the difference is +0.012 F1 [+0.005, +0.019] and +0.060 TEDS [+0.046, +0.074].

The combination is thus better than either source alone. Because it was designed after the first test results were known, we regard it as evidence that the architecture can absorb a second observer, not as a pre-registered result.

5.4 Content Preservation and Cost

Table 3 and Fig. 3 compare generative rewriting with anchor routing on the same documents, using one measure for both. They answer the question of Sect. 1.

Heading accuracy

On these papers the two approaches are not significantly different. The full system's difference is −0.014 F1 [−0.056, +0.030] and −0.027 TEDS [−0.072, +0.019]; this is not evidence of equivalence. Measured from the deterministic core (0.806 F1, 0.681 TEDS on these documents), anchor routing keeps 86% of the generative model's F1 gain (95% interval 54%–157%) and 83% of its TEDS gain (59%–114%); the intervals are wide because the generative gain itself is small on 100 documents.

Preservation

The generative output contains 96.7% of the source words on average, and 69.2% on its worst document; none of its outputs reached the output limit. It preserves fewer words than Constellation on 67 of the 100 documents, and 1.1% of its output words do not come from the source. Constellation's export contains 99.3% of the source words (86.3% on its worst document). Its export contains every block whatever the votes; the deterministic export of the same documents gives the same mean (0.993). Its residual loss arises in text extraction, not in assembly: on all 454 test documents the export contains every text block of the provider in order (minimum 1.000). The words that the provider does not export are mostly glyphs drawn inside figures, repeated running heads, which it sets aside as page decoration, and some formula symbols. Its worst document is a paper whose plot markers are text glyphs. Its 1.0% of unmatched output words are word fragments where the provider splits or joins words differently from PyMuPDF, not added content; markup such as `<br>` in table cells is ignored for both systems. In-order preservation is slightly lower for Constellation (0.953 vs 0.960) because the provider reorders columns and formula fragments, whereas the generative baseline copies the order of the PyMuPDF text it was given.

Cost

Input tokens are similar (11,893 vs 12,030), because these papers are short enough for the control view to cover most of them. The generative output, however, is 19× larger than Constellation's anchors, and a document takes 2.2–3.6× as long (range over the four runs of the full system).

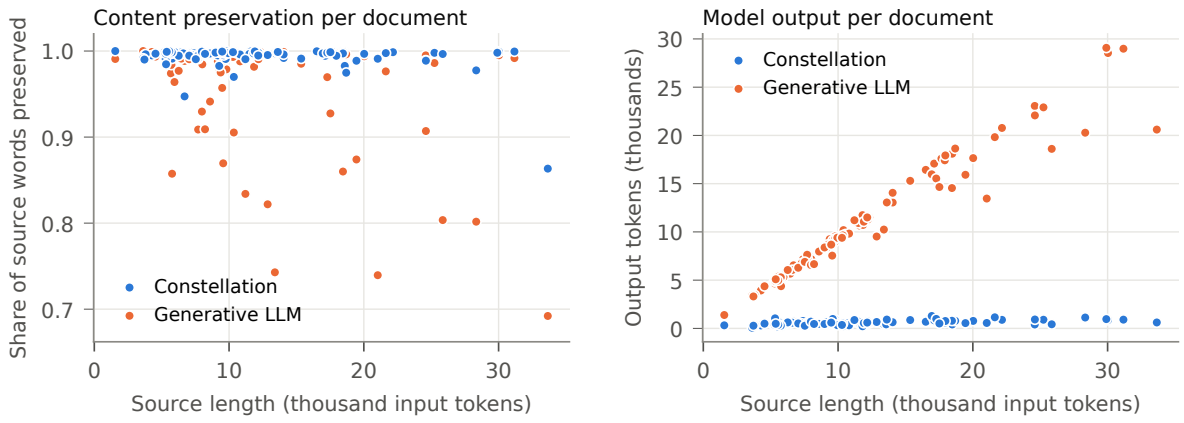

Fig. 3 Generative rewriting versus anchor routing on the same 100 test documents, against source length. Left: share of the words of the PDF text layer present in the exported Markdown (Constellation: the repeat run that stored its export). Right: model output tokens per document (Constellation: main run).

Wall times were measured while other runs shared the machine and the gateway, so they are indicative only. The generative baseline also dropped the page markers in 24 outputs, which we repaired for scoring.

5.5 Controlled Typography and Long Documents

The three suites of this section (mis-styled DOCX, unstyled PDFs, long documents) come from the repository's acceptance tests and were used during development. They are therefore regression checks, not held-out evaluations. They are scored on the frozen code with exact title matching after case folding, whitespace removal and number stripping.

Mis-styled DOCX

Twenty-four DOCX files render three logical structures under eight typographic conditions:

- correct styles;
- styles shifted one level;
- a uniform style;
- styled prose;
- styled metadata lines;
- top-level numbering written as "1.1";
- deeper headings set larger;
- everything bold.

The deterministic core reaches F1 1.000 and the full system 1.000, so this suite only shows that the typed evidence handles these conditions without the reader.

Unstyled PDFs

On four synthetic PDFs set in one font with no bold or size cues, the deterministic core recovers 0.745 of the headings (F1 0.854) and the full system 1.000 (F1 1.000). Here the reader supplies the semantic judgement that typography cannot.

Long documents

Twelve long PDFs (papers, surveys, a monograph and a standard; 255–4,115 blocks) exercise the budget and sharding. The production control view is on average 81% smaller than the full block text, and the full system issues 9.6 requests per document on average. Their references derive from the embedded outlines, which were visible to the system and list only the first two or three levels; deeper genuine headings therefore count as false positives. We use this suite for cost and scaling only.

5.6 Error Analysis

Fig. 4 shows the typical effect of the reader. We also drew a seeded random sample of the false positives of the full system (no outline) on the test split. Of the sampled items, 98 remain false positives under the final title convention; they form a uniform sample of the 758 final false positives. Each was judged from the predicted line, its

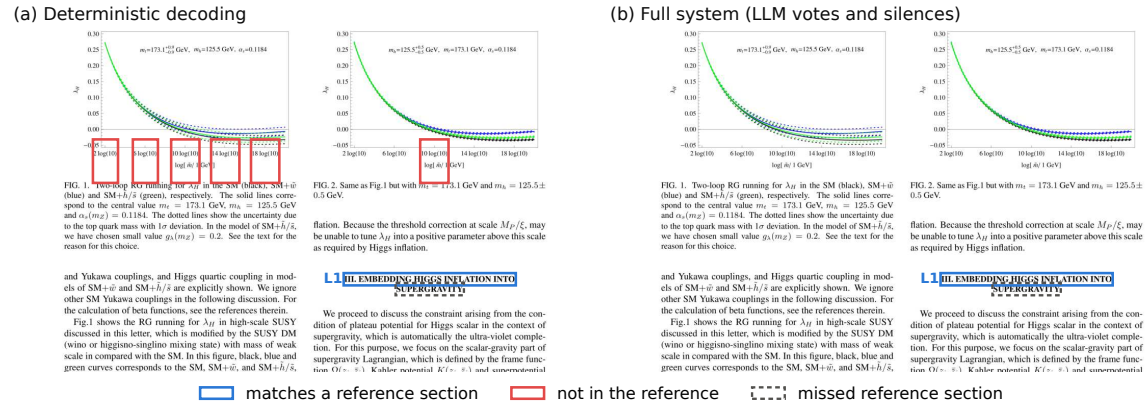

Fig. 4 Second page of a development document (arXiv 1504.08093). (a) Deterministic decoding accepts six axis-tick labels inside the plots as headings: they are short, isolated lines that no negative evidence covers. (b) The LLM reader was shown these candidates and did not vote for them, and the silence channel removes them; the section heading is kept at its level. The dashed box marks the second line of that heading, which the reference annotates as a separate section; both systems merge it into the title and are counted as missing it.

neighbouring lines with their typography, and the nearby reference sections:

- 26% are genuine section titles that the reference omits: front- and back-matter titles (Abstract, Acknowledgments, References) and numbered subsections, including run-in ones;
- 5% belong to referenced sections whose title is split across lines or whose page or wording differs in the fetched arXiv version;
- the remaining 69% are true errors: pseudo-code and displayed-formula lines (21), text inside figures and tables (18), bold run-in labels, theorem labels and list items (18), title-page lines or running headers (4), bibliography entries (4) and bold body lines (3).

Reported precision is therefore a lower bound for the audited system, and plausibly for every system that detects run-in numbered subsections, which the reference includes for some documents and omits for others; we did not alter the reference. The judgements were made by one AI-assisted annotator. The author reviewed a seeded sample of 20 of them, with the AI verdict visible, and agreed with 20 of the 20; this is a consistency check, not an independent agreement study. Every item-level verdict and the author's review are archived with the results.

Missed sections in the inspected documents fall mainly into three cases:

- run-in headings that begin a paragraph;
- multi-line titles that the reference splits into separate lines;
- italic IEEE-style third-level subsections set in the body font.

6 Discussion

What the LLM contributes

The reader's main effect is to veto proposals. Deterministic evidence is tuned for recall, because a missed heading merges two sections, whereas a false heading only splits one; without a model, precision is the bottleneck. Treating silence as an observation, with a likelihood ratio fixed from two interpretable rates, turns the reader into a precision filter without letting it move or rewrite content. On these papers its heading accuracy is not significantly different from that of a generative rewrite, at a fraction of its output.

Observers, not pipelines

The layout variant applies the same idea to a second observer. On this benchmark the detector finds headings more reliably than the text-only system but assigns no depth, and the reader still adds +0.021 F1 on top of the detector's evidence. Fusing both sets of detections and silences in one decoder outperformed each source alone. Because

the detector's heading regions become typed evidence, the content guarantee is unaffected.

Scope and limitations

- The method targets born-digital DOCX/PDF; scanned pages require upstream OCR.
- Run-in headings are detected as title fragments of a paragraph, but the benchmark annotates them inconsistently and our matching is line-based.
- Evidence likelihood ratios and the decoder's scoring rule are versioned heuristics, not calibrated probabilities. The reader rates behind λ_{sil} ($r = 0.9$, $f = 0.18$) were assumed, not measured, and λ_{sil} was not swept.
- The decoder's components and the routing budget were not ablated on Comp-HRDoc, where the control view covers most of each paper.
- DOCX handling is tested only on the synthetic development suite of Sect. 5.5; no held-out DOCX benchmark was used.
- Matching tolerates prefix agreement; the strict rule lowers all systems by similar amounts but makes the top two systems tie on F1.
- The evaluation uses one LLM. Other models may differ in recall and false-vote rate, which would change the appropriate λ_{sil}.
- Three analyses were added after the first test results: the layout variant, the Docling + numbering baseline and the strict-matching analysis. The variant's ratios were fixed on development data, but the comparison is not pre-registered.
- The PDFs were fetched from arXiv after the benchmark's release. The version check removes 15 documents, but small differences between versions may remain.
- Comp-HRDoc covers short scientific papers, on which the sparse view saves little input. Reports, manuals and multilingual office documents without outlines remain to be evaluated at scale.

7 Conclusion

Constellation lets an LLM decide where sections start without letting it carry the document. Typed deterministic evidence proposes, the model votes or stays silent over a sparse view, a constrained decoder fuses both, and closure copies the original blocks into disjoint sections. On 454 Comp-HRDoc test documents with the outline hidden, the reader raises F1 from 0.776 to 0.881 and TEDS from 0.644 to 0.791 over deterministic decoding. On a 100-document sample its heading accuracy is not significantly different from generative rewriting, while it emits 19× fewer output tokens and its export preserves 99.3% of the source words against 96.7%. In a post-hoc variant, adding a layout detector's detections and silences as evidence yields the best system in our comparison (0.937 F1, 0.865 TEDS).

Statements and Declarations

Funding. No funding was received for this work.
Competing interests. The author declares no competing interests.
Ethics approval and consent to participate. Not applicable.
Data availability. Comp-HRDoc annotations are available from https://github.com/microsoft/CompHRDoc (MIT licence); the PDFs are public arXiv documents. Document identifiers, SHA-256 hashes of the fetched PDFs, the admission list, the development/test split and all per-document predictions and scores are archived on Zenodo with the code (https://doi.org/10.5281/zenodo.23358153).
Code availability. Source code, the evaluation scripts (`evaluation/hrdoc`) and the scripts that generate every reported number are available at https://github.com/1911342723/Constellation and archived on Zenodo (https://doi.org/10.5281/zenodo.23358153, tag `ijdar-submission-v2`). The evaluated versions are tagged `ijdar-frozen` (frozen code), `ijdar-perf` (exact performance fix) and `ijdar-layout` (layout evidence), as described in Sect. 4.1. Superseded runs are kept with the results.
Use of AI tools. Large language models are a component of the studied method. Separately, AI tools were used as assistants in writing parts of the code and in drafting and editing the text, and in the error audit of Sect. 5.6, as described there. The author designed the method, wrote most of the code, revised the manuscript and takes full responsibility for its content.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474 (2020)

[2] Apache Software Foundation: Apache Tika. https://tika.apache.org/. Accessed 2026-06-09

[3] Microsoft: MarkItDown. https://github.com/microsoft/markitdown. Accessed 2026-06-09

[4] Williamson, M.: Mammoth: Convert .docx Documents to HTML. https://github.com/mwilliamson/python-mammoth. Accessed 2026-06-09

[5] Zhong, X., Tang, J., Jimeno Yepes, A.: PubLayNet: Largest dataset ever for document layout analysis. In: International Conference on Document Analysis and Recognition (ICDAR), pp. 1015–1022 (2019). https://doi.org/10.1109/ICDAR.2019.00166

[6] Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A.S., Staar, P.J.: DocLayNet: A large human-annotated dataset for document-layout segmentation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 3743–3751 (2022). https://doi.org/10.1145/3534678.3539043

[7] Auer, C., Lysak, M., Nassar, A., Dolfi, M., Livathinos, N., Vagenas, P., Ramis, C.B., Omenetti, M., Lindlbauer, F., Dinkla, K., Mishra, L., Kim, Y., Gupta, S., Lima, R.T., Weber, V., Morin, L., Meijer, I., Kuropiatnyk, V., Staar, P.W.J.: Docling technical report. Technical report, IBM Research (2024). arXiv:2408.09869; software version 2.138.0 used in this study, https://github.com/docling-project/docling

[8] Ma, J., Du, J., Hu, P., Zhang, Z., Zhang, J., Zhu, H., Liu, C.: HRDoc: Dataset and baseline method toward hierarchical reconstruction of document structures. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 1870–1877 (2023). https://doi.org/10.1609/aaai.v37i2.25277

[9] Wang, J., Hu, K., Zhong, Z., Sun, L., Huo, Q.: Detect-Order-Construct: A tree construction based approach for hierarchical document structure analysis. Pattern Recognition **156**, 110836 (2024) https://doi.org/10.1016/j.patcog.2024.110836

[10] Xing, H., Cheng, C., Gao, F., Shao, Z., Yu, Z., Bu, J., Zheng, Q., Yao, C.: DocHieNet: A large and diverse dataset for document hierarchy parsing. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 1129–1142 (2024). https://doi.org/10.18653/v1/2024.emnlp-main.65

[11] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.*: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)

[12] OpenAI: GPT-4 technical report. arXiv:2303.08774 (2023)

[13] Doucet, A., Kazai, G., Colutto, S., Mühlberger, G.: ICDAR 2013 competition on book structure extraction. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1438–1443 (2013). https://doi.org/10.1109/ICDAR.2013.290

[14] Juge, R., Bentabet, I., Ferradans, S.: The FinTOC-2019 shared task: Financial document structure extraction. In: Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019), pp. 51–57. Linköping University Electronic Press, Turku, Finland (2019). https://aclanthology.org/W19-6407/

[15] Lopez, P.: GROBID: Combining automatic bibliographic data recognition and term

extraction for scholarship publications. In: Research and Advanced Technology for Digital Libraries (ECDL), pp. 473–474. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04346-8_62

[16] Wang, B., Xu, C., Zhao, X., Ouyang, L., Wu, F., Zhao, Z., Xu, R., Liu, K., Qu, Y., Shang, F., Zhang, B., Wei, L., Sui, Z., Li, W., Shi, B., Qiao, Y., Lin, D., He, C.: MinerU: An open-source solution for precise document content extraction. arXiv:2409.18839 (2024)

[17] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: LayoutLM: Pre-training of text and layout for document image understanding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1192–1200 (2020). https://doi.org/10.1145/3394486.3403172

[18] Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.: LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 2579–2591 (2021). https://doi.org/10.18653/v1/2021.acl-long.201

[19] Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: Pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 4083–4091 (2022). https://doi.org/10.1145/3503161.3548112

[20] Shen, Z., Zhang, R., Dell, M., Lee, B.C.G., Carlson, J., Li, W.: LayoutParser: A unified toolkit for deep learning based document image analysis. In: Document Analysis and Recognition – ICDAR 2021, pp. 131–146. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86549-8_9

[21] Blecher, L., Cucurull, G., Scialom, T., Stojnic, R.: Nougat: Neural optical understanding for academic documents. arXiv:2308.13418 (2023)

[22] Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: Data, model, and evaluation. In: Computer Vision – ECCV 2020. Lecture Notes in Computer Science, vol. 12366, pp. 564–580. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_34

[23] Pawlik, M., Augsten, N.: Tree edit distance: Robust and memory-efficient. Information Systems **56**, 157–173 (2016) https://doi.org/10.1016/j.is.2015.08.004