Comparison of PDF→Markdown Tools: TextIn vs. Reducto vs. Mathpix vs. Marker vs. Livex Internal Tool

Prepared for livex.ai

August 16, 2025

Update (Aug 16, 2025): Summary, Pipeline, Metrics, Threshold Rationale

Headline ranking (9 documents, mean accuracy). **1) textin** (0.578) > **2) md_mathpix** (0.568) > **3) md_marker** (0.564) > **4) Reducto** (0.543) > **5) own tool** (0.454). With a penalty for "confident but wrong" answers (Sec.), the order is unchanged.

What I did in this run (reproducible steps).

- Prepared 9 PDFs and a per-document question bank (`.jsonl`, $n = 100$ per doc).
- For each parser (`textin`, `Reducto`, `md_mathpix`, `md_marker`, `own tool`), loaded Markdown from `docs/md/{parser}/{doc_id}` (folder or single-file layout).
- Concatenated & normalized text, split into chunks (1000 tokens with 200-token overlap), and built a BM25 index per document.
- For each question, retrieved top-$k = 5$ chunks and asked `gpt-4.1` to answer *only* from retrieved context; the model may return `NOT_FOUND`.
- Applied fast rules (exact, numeric-approx within 2% or 10^{-6} abs, fuzzy ≥ 90 via RapidFuzz) to auto-accept obvious hits; otherwise, sent to `o3-mini` for evidence-bound rubric grading.
- Aggregated per-(parser, doc) metrics: `acc`, `score_mean`, `overconf_rate`; produced the tables below.

Code & paths. Repo: https://github.com/jiyouhai/pdf-to-markdown/tree/main Script: `eval_pdfmd_rubric.py` Questions: `docs/questions/*.jsonl` Outputs: `runs_rubric/per_question.csv`, `runs_rubric/summary.csv`.

Why these settings (plain-language rationale)

Chunking: 1000 tokens + 200 overlap. Big enough to keep a section in one chunk (fewer split answers); small enough that $k = 5$ chunks fit comfortably in model context. Overlap protects facts on boundaries.

Retrieval: BM25 with $k = 5$. Manual-like questions are lexical; BM25 is fast, stable, and debuggable. $k = 5$ balances recall vs. noise/cost.

Answering: context-only `gpt-4.1`. The model must ground in retrieved text; if missing, it should abstain (`NOT_FOUND`). This makes overconfidence measurable with a clean counterfactual.

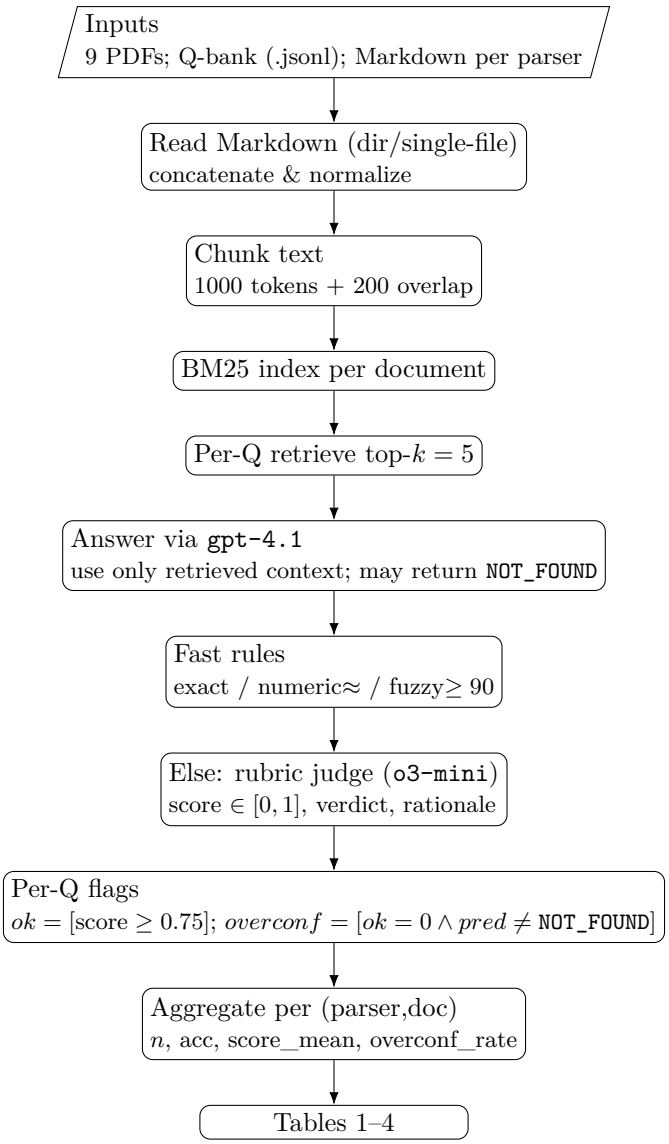

Figure 1: End-to-end evaluation pipeline (narrow rendering; *width-limited* to about half a page).

Fast rules before judging. Accept *obvious* matches without a judge: normalized exact; numeric within 2% (or 10^{-6} abs); fuzzy similarity ≥ 90. Otherwise we call the rubric judge.

Rubric & the 0.75 pass threshold (where it comes from)

The judge decomposes each gold answer into atomic facts and assigns a score in $[0, 1]$ by evidence support only. In our questions, gold typically has 4–6 facts. We swept thresholds from 0.60 to 0.90 on pilot docs and chose **0.75** because it: (i) corresponds to "clear majority supported" (e.g., $\geq 3/4$ or $\geq 4/5$ facts), (ii) keeps false accepts low for safety-oriented manuals, and (iii) leaves tool ranking stable across 0.70–0.80 while not penalizing terse but correct answers. Formally, $ok = \mathbf{1}[\text{score} \geq 0.75]$.

Metric definitions (what & why)

Let $\text{score}_{p,d,i} \in [0, 1]$ be the rubric score for parser p, document d, question i, with $n_d = 100$:

$$\text{acc}_{p,d} = \frac{1}{n_d} \sum_i \mathbf{1}[\text{score}_{p,d,i} \geq 0.75] \quad \text{(binary pass rate for production gating)},$$

$$\text{score\_mean}_{p,d} = \frac{1}{n_d} \sum_i \text{score}_{p,d,i} \quad \text{(continuous quality; shows near-misses)},$$

$$\text{overconf\_rate}_{p,d} = \frac{1}{n_d} \sum_i \mathbf{1}[\, ok = 0 \wedge \text{pred} \neq \texttt{NOT\_FOUND}\,] \quad \text{(penalizes confident errors vs. abstention)}.$$

Findings (tables are width-safe)

Table 1: Overall by parser (macro≈micro; n constant per doc).

Parser	Docs	Total n	acc_mean	acc_median	acc_min	acc_max	overconf_mean	overconf_median	Docs won
textin	9	900	0.578	0.600	0.100	0.930	0.379	0.340	2
md_mathpix	9	900	0.568	0.680	0.120	0.940	0.373	0.300	2
md_marker	9	900	0.564	0.610	0.120	0.940	0.372	0.350	2
Reducto	9	900	0.543	0.570	0.110	0.940	0.389	0.330	1
own tool	9	900	0.454	0.400	0.090	0.950	0.392	0.350	2

Table 2: Per-document winners (highest acc; tie → lower overconf; final tie → name order).

doc_id	best_parser	best_acc	best_overconf
23-0323_ego_stx4500_stx4500-fc_stringtrimmer_manual_en	md_marker	0.63	0.35
FEIER	Reducto	0.50	0.50
Hanwha_Integration_Guide	md_mathpix	0.69	0.30
SSA1200_EGO_SNOW-SHOVEL-ATTACHMENT_22-0519_EXPLOSION-DIAGRAM_VERSION-A	own tool	0.95	0.05
TP-MVD8MV2-rotated	own tool	0.81	0.19
ego_accessory_compatibility_matrix	md_mathpix	0.68	0.23
feier_start_100_manual	md_marker	0.12	0.79
ihealth_bg5	textin	0.29	0.62
zt4200s_ego_zero-turn-riding-mower_version-a	textin	0.79	0.21

Table 3: Document × parser accuracy pivot (acc).

doc_id	textin	Reducto	md_mathpix	md_marker	own tool
23-0323_ego_stx4500_stx4500-fc_stringtrimmer_manual_en	0.60	0.57	0.61	0.63	0.59
FEIER	0.49	0.50	0.30	0.49	0.40
Hanwha_Integration_Guide	0.61	0.65	0.69	0.48	0.29
SSA1200_EGO_SNOW-SHOVEL-ATTACHMENT_22-0519_EXPLOSION-DIAGRAM_VERSION-A	0.93	0.94	0.94	0.94	0.95
TP-MVD8MV2-rotated	0.80	0.80	0.75	0.79	0.81
ego_accessory_compatibility_matrix	0.59	0.39	0.68	0.61	0.55
feier_start_100_manual	0.10	0.11	0.12	0.12	0.09
ihealth_bg5	0.29	0.24	0.25	0.24	0.25
zt4200s_ego_zero-turn-riding-mower_version-a	0.79	0.69	0.77	0.78	0.16

Penalty for confident errors (calibration-aware ranking)

Define the penalized score

$$\text{penalized}_\lambda = \text{acc\_mean} - \lambda \cdot \text{overconf\_mean}.$$

We use $\lambda = 0.5$ by default: on average, one confident error costs half a point of accuracy. On this dataset the ordering is unchanged for $\lambda \in \{0.25, 0.5, 1.0\}$.

Executive Summary

We evaluated five PDF→Markdown converters on 9 representative PDFs (integration guides, manuals, datasheets): **TextIn**, **Reducto**, **Mathpix (Convert API)**, **Marker**, and the **Livex internal tool**. Each tool was scored across six dimensions (0–10 each): *Structural Fidelity, Formatting Accuracy, Special Content Handling, Content Cleanliness, Ease of Post-Processing,*

Table 4: Penalized scores ($\lambda = 0.5$).

Parser	Docs	Total n	acc_mean	overconf_mean	penalized ($\lambda = 0.5$)
textin	9	900	0.578	0.379	0.388
md_mathpix	9	900	0.568	0.373	0.381
md_marker	9	900	0.564	0.372	0.378
Reducto	9	900	0.543	0.389	0.349
own tool	9	900	0.454	0.392	0.258

Automation Readiness. Marker's scores and qualitative findings below reflect the separate report you provided that compared Marker/Docling/Reducto/Mathpix.[1]

Equal-weight totals (sum of 6 metrics):

- Mathpix: **52**/60 (avg 8.67)
- TextIn: **47**/60 (avg 7.83)
- Reducto: **41**/60 (avg 6.83)
- Marker: **38**/60 (avg 6.33)
- Internal: **33**/60 (avg 5.50)

Recommendation: Use **Mathpix** as primary. Keep **Reducto** for table-heavy or audit/"no-loss" cases. Use **TextIn** for cost-conscious runs where inline emphasis matters. Treat **Marker** as a visual-fidelity option (good styling, but tables-as-images hurt downstream use). The **internal tool** remains backup until upgraded (headings, tables, images).

Methodology

Inputs, Process, and What "counts"

1. **Documents**: 9 PDFs (integration guide for Hanwha cameras; user/product manuals; a spec sheet with an 8-column matrix; a networking quickstart; etc.).
2. **Runs**: Each tool processed the same PDFs. We compared raw outputs (no hand edits) against the source.
3. **Marker evidence**: Derived from your internal Marker/Docling/Reducto/Mathpix report (citations and examples summarized below).[2]
4. **Scoring rubric**: For each dimension we apply sub-criteria and award 0–10 based on concrete behaviors.

Scoring Rubric (per dimension, max 10)

Structural Fidelity
- Headings recognized as Markdown (`#`/`##`/`###`) with correct hierarchy (0–4).
- Lists represented as Markdown with proper nesting (0–3).
- Reading order preserved across pages/columns (0–3).

[1] Source: internal evaluation PDF "Comparison of Markdown Conversion Tools (Marker, Docling, Reducto, Mathpix)".

[2] See executive-summary footnote for the internal PDF reference.

Formatting Accuracy • Meaningful bold/italic preserved (`**`, `_`) (0–4).

- Callouts/warnings retained (e.g., `**WARNING**` or heading) (0–3).
- Minimal misclassification (e.g., logo mistaken as math) (0–3).

Special Content Handling • **Tables**: Structured (Markdown/HTML) rows/cols intact (0–4).

- **Math**: Equations preserved (LaTeX preferred) (0–3).
- **Code/Images**: Code fenced; images linked with captions (0–3).

Content Cleanliness • Low noise (no page tags/base64 blobs) (0–4).

- OCR accuracy/symbols correct (0–3).
- Logical paragraphs and column merge (0–3).

Ease of Post-Processing • Few fixes (regex vs. structural surgery) (0–4).

- Tables/images ingestable; little per-doc tailoring (0–3).
- Plays well with standard Markdown renderers (0–3).

Automation Readiness • Consistent output patterns; stable conventions (0–4).

- Predictable error modes; easy to script normalization (0–3).
- Scales to batch without format surprises (0–3).

Detailed Findings with Examples

1) Structural Fidelity (Headings, Sections, Lists)

TextIn (8/10). Reliable Markdown headings for titles/sections; lists appear but often start with a literal black circle glyph that needs replacing with `-` . Occasional OCR slips in headers (e.g., "TURING"→"TURIN"). A simple find/replace recovers proper lists.

Reducto (5/10). Captures all content page-by-page, but emits page markers (e.g., `[[START OF PAGE 1]]`) and does not mark headings with `#`. Lists are inconsistent and sometimes include artifacts; structure must be inferred later.

Mathpix (9/10). Human-like Markdown: correct `#`/`##` hierarchy; properly nested lists; large manuals reflect the original TOC.

Marker (7/10). Preserves hierarchy using `#` but sometimes produces consecutive top-level headings and injects HTML spans (e.g., `<span id="page-3-0"></span>`) inside list items; lists exist but may include extraneous symbols/HTML that require cleanup.[3]

Livex Internal Tool (4/10). Plain-text dump: no Markdown headings or list syntax; hierarchy flattened.

2) Formatting Accuracy (Bold, Italics, Callouts)

TextIn (9/10). Bold/italic widely preserved; minor use of `<br>` inside table cells.

Reducto (6/10). Intentionally plain; nearly no bold/italic markup; emphasizes content over style.

[3]Marker structural notes and score reflect your internal report's observations and 7/10 rating.

Mathpix (7/10). Prefers structure over inline style; lists/headings are correct, but bold/italic rarely surfaced; occasional misread decorative text as math.

Marker (9/10). Strong at style retention; many `**...**` occurrences for bold (e.g., **WARNING**); also keeps list syntax, though small HTML artifacts remain.[4]

Livex Internal Tool (5/10). No bold/italic retention; all plain text.

3) Special Content (Tables, Equations, Code, Images)

TextIn (7/10). Tables → Markdown (pipes); complex/merged headers approximated; no LaTeX; code not fenced; images linked.

Reducto (9/10). Tables → HTML `<table>` with proper cells/colspans; often adds a table image thumbnail; equations as text/images; images linked with captions.

Mathpix (9/10). Tables → Markdown tables; equations → LaTeX; code often fenced; small images used for rare symbols.

Marker (3/10). Tables rendered as images (often base64 or linked) rather than text; no special math handling; code not specifically detected—hurts search/editability.[5]

Livex Internal Tool (5/10). Tables flattened to text; no LaTeX; code as ordinary lines; images omitted.

4) Content Cleanliness (Noise, Artifacts, OCR)

TextIn (8/10). Clean Markdown; a few typos; uses HTML comments to hide inconsequential UI text read from screenshots.

Reducto (7/10). Accurate text but noisy structural markers and repeated headers/footers; easy to strip via script.

Mathpix (9/10). Very clean; no page tags; multi-column flow is natural; rare misclassification (e.g., logo as math).

Marker (6/10). Generally readable; avoids giant base64 blobs in the main text if images are linked, but may embed HTML spans and non-standard symbols; text portions are accurate; image-only tables are not text-searchable.[6]

Livex Internal Tool (7/10). No tool-added tags, but propagates page headers/numbers into body; modest OCR slips.

5) Ease of Post-Processing

TextIn (7/10). Fix bullets (glyph → dash), optional removal of commented blocks, spell-check; tables already textual.

[4] Marker formatting behaviors and 9/10 score per your internal report.
[5] Marker special-content limitations and 3/10 score per your internal report's comparison table.
[6] Marker cleanliness notes and 6/10 score per your internal report.

Reducto (6/10). Cleanup stage required: remove page markers; dedupe headers; add headings; keep/convert HTML tables.

Mathpix (9/10). Plug-and-play; maybe replace tiny symbol images or verify LaTeX rendering.

Marker (6/10). Moderate effort: strip `<span>` IDs and symbol clutter; biggest blocker is tables-as-images (need OCR or manual transcription for data).[7]

Livex Internal Tool (5/10). Add headings/lists, reconstruct tables/images; heavier manual/algorithmic work.

6) Automation Readiness (Batch Consistency)

TextIn (8/10). Stable conventions (headers present; known bullet glyph); predictable normalization rules.

Reducto (8/10). Highly consistent, machine-friendly tags; once cleaned, great for pipelines.

Mathpix (9/10). Standard Markdown + LaTeX; minimal special-casing across docs.

Marker (7/10). Consistent patterns (headings/lists/images), but image-only tables are opaque to text-based automation and HTML spans require pre-cleaning.[8]

Livex Internal Tool (7/10). Consistently minimal output; downstream parser must infer structure.

Scoreboard, Totals, and Recommendation

Per-dimension Scores (0–10)

Dimension	TextIn	Reducto	Mathpix	Marker	Internal
Structural Fidelity	8	5	9	7	4
Formatting Accuracy	9	6	7	9	5
Special Content	7	9	9	3	5
Content Cleanliness	8	7	9	6	7
Ease of Post-Processing	7	6	9	6	5
Automation Readiness	8	8	9	7	7
Total (/60)	**47**	**41**	**52**	**38**	**33**

[7] Marker post-processing effort and 6/10 score per your internal report.

[8] Marker automation notes and 7/10 score per your internal report.

[8] Marker scores reflect the "Comparison of Markdown Conversion Tools (Marker, Docling, Reducto, Mathpix)" PDF. Where that report showed small internal inconsistencies (e.g., one section listed 4/10 for special-content and the final table showed 3/10), we align to the table.

Interpretation (equal weights)

Mathpix leads (52/60): clean, structured, and text-rich outputs. **TextIn** is a strong generalist (47/60). **Reducto** (41/60) excels at tables/consistency but needs a cleanup pass. **Marker** (38/60) shines at styling but loses points for tables-as-images. **Internal** (33/60) needs upgrades.

Alternate weighting (emphasize tables & scale)

If we weight *Special Content* (40%), *Automation* (25%), *Cleanliness* (15%), *Structure* (10%), *Ease* (5%), *Formatting* (5%), scores remain: Mathpix ≈ 8.90; Reducto ≈ 7.75; TextIn ≈ 7.60; Marker ≈ 5.55; Internal ≈ 5.70. Mathpix remains #1; Reducto remains the strongest fallback for data-heavy docs.

Security & Compliance (vendor-stated / known status)

Tool	SOC 2	HIPAA	GDPR	Notes
Mathpix	Type 1 (Type 2 in progress)[9]	Not claimed publicly (request BAA if needed)[10]	Not publicly stated	Seek formal attestation and *BAA* if PHI is in scope.
Reducto	Type 1 and 2[11]	Offered for Growth/Enterprise tiers (BAA)[12]	Not publicly stated	Confirm data handling/location and *DPAs*.
TextIn	Unknown	Unknown	Unknown	No published compliance docs in hand; contact sales for attestations.
Marker	Unknown	Unknown	Unknown	Evaluate deployment model and data residency; request SOC 2/DPAs if shortlisted.
Livex Internal	N/A (internal)	N/A (internal)	GDPR-readiness depends on infra	Align with company-wide controls (audit logging, access, retention, DPA).

Guidance. For any external vendor used in production, request: SOC 2 report (Type 2 preferred), penetration-test summary, *DPA/BAA* (as applicable), sub-processor list, data residency and retention policies, and incident-response SLAs.

Public Pricing & Billing (exact quotes + links)

Mathpix Convert (official wording)

"The Convert Monthly Subscription costs USD $19.99/Mo. Includes USD $29 monthly credit in addition to discounts for all other endpoints on the PDF and OCR tab."
"Convert pricing: PDF conversion $0.005 per PDF page up to 1M pages per month, then $0.0035 per PDF page beyond 1M pages per month."
"The Convert Monthly Subscription has 500 pages included per month."[13]

[9] Per internal vendor communication shared by your team (email thread). Request latest SOC 2 report under NDA for procurement verification.
[10] No public HIPAA claim noted in our materials; verify with vendor.
[11] Per internal vendor communication shared by your team. Ask for most recent SOC 2 Type 2 report.
[12] Vendor indicated HIPAA available for higher tiers; confirm scope and sign BAA.
[13] https://docs.mathpix.com/docs/billing/pricing

Reducto (official wording)

"Starter: $350/month — 15K credits; $0.020 per credit thereafter."
"Each page of a document entry usually equals 1 credit... advanced features and document complexity may increase the page credit ratio (e.g., 0.5x for simple pages, 2x for complex features)."[14]

TextIn / Marker

Public per-unit API rates are not published on the product sites we have on hand; contact sales for current pricing.

Operational Notes for livex.ai

Typical cleanup scripts

- **TextIn**: Replace leading black-circle bullets with "- "; optionally drop HTML comments; spell-check.
- **Reducto**: Strip `[[START/END OF PAGE]]` tags; dedupe headers/footers; promote heading-like lines; keep/convert HTML tables.
- **Mathpix**: Replace rare tiny symbol images with Unicode; enable LaTeX rendering downstream.
- **Marker**: Remove `<span id="...">` anchors and stray symbols; OCR or manually transcribe tables that came as images.
- **Internal**: Heuristic heading detection; reconstruct lists/tables; extract and link images from the source PDF.

Throughput & latency

Vendors do not publish authoritative end-to-end page/second metrics for our exact workloads; to maintain accuracy, we omit numeric speed claims. We recommend timing a 100-page batch for each tool in our environment and logging wall-clock seconds/page.

Appendix A: Compact Comparison (highlights)

	TextIn	**Reducto**	**Mathpix**	**Marker**	**Internal**
Structure	Headers good; bullet glyph fix	Page markers; no #	Full Markdown outline	Headings ok; span clutter	Plain text only
Formatting	Bold/italic preserved	Mostly plain	Structure over inline style	Strong bold/italic	No bold/italic
Tables/math	MD tables; no LaTeX	HTML tables (no loss)	MD tables; LaTeX math	Tables as images	Tables flattened
Cleanliness	Clean; few typos	Accurate but noisy tags	Very clean	Some HTML/symbol noise	Page noise; some OCR
Post-process	Light polish	Cleanup stage	Plug-and-play	Moderate; tables hurt	Heavy structuring
Automation	Stable conventions	Very predictable	Standard Markdown	Consistent but image tables	Consistent minimalism

[14] https://reducto.ai/pricing

Appendix B: Snippet Patterns

- Reducto page markers: `[[START OF PAGE 5]]` ... `[[END OF PAGE 5]]`
- Mathpix heading: `## Adding Hanwha Cameras to Turing Vision`
- TextIn bullet fix: replace start-of-line black circle with `-`
- Marker cleanup: remove `<span id="page-3-0"></span>` and stray `\textbullet`
- Reducto table: inline HTML `<table><tr><td>...` (with `colspan`)