
Arabic Text from PDFs: Four of Eight Extractors Reverse the Lam–Alef Ligature, a Fifth Once Reordered

Dmitrii Lozhkin

AVELIN AI · October 26, 2026

Abstract

Arabic retrieval systems usually index the text layer of born-digital PDFs, and a common integrity check confirms that every character and number survives extraction. That check cannot see order. We extracted 259 pages of five public Arabic PDFs with eight open-source engines. Four of them (Poppler, MuPDF, PDFium, pdf.js) wrote the lam–alef ligature backwards, and pdfminer’s display-order output did once we restored reading order; PDFBox and docling-parse kept it. A copy with only the order inside multi-character font mappings reversed made these five correct and broke the two: these engines reverse right-to-left text one code point at a time, and the ligature with it. In Poppler’s output 8.9% of 126,504 Arabic words changed, with identical character counts on every page. A regular expression for a letter sequence no Arabic word contains flagged 204 of 208 text pages and no clean text. Tesseract OCR read the ligature but lost or misread 4.8% of words. Top-four hits of BM25 and dense retrieval did not change on 12 questions, though ranks moved; exact matching and quoted text fail.

1 Introduction

PDF extraction can misidentify words, misorder them or join paragraphs wrongly; Bast and Korzen scored 14 tools on 12,098 articles by these criteria [1]. Arabic adds joined letters, right-to-left order and obligatory ligatures. On KITAB-Bench the best system reached 65% accuracy in Arabic PDF-to-Markdown conversion [2]; on MDPBench open-source parsers lose 14.0% on non-Latin scripts [3].

Pipelines that read the text layer instead meet a failure known to practitioners: the ligature ﻻ (lam + alef), drawn as one glyph, comes back as ﻻ. It is reported for pdf.js [4] and Poppler [5], both open without maintainer response, and was declined for PyMuPDF [6]. We found no measurement of its prevalence, damage, detection or retrieval cost; we measure all four with eight engines and OCR, test the mechanism by editing only the font mappings, and give a one-line detector.

2 Setup

Documents. Five public Arabic PDFs from four reports of international scientific bodies, 259 pages, all produced with Adobe InDesign. Their fonts carry ToUnicode maps, which tell an extractor which characters a glyph stands for. A *word* is a maximal run of Arabic letters after NFKC (Unicode compatibility) normalization, without short-vowel marks and tatweel (the stretching stroke).

Diagnostic copy. In a copy of each PDF, every ToUnicode entry mapping one glyph to two or more Arabic code points lists them in reverse order: 129 entries, 86 of them lam–alef, padded so that no other byte changes. An engine that the copy repairs reverses multi-character glyphs; one that it breaks keeps the font’s order.

Engines. Thirteen configurations of the eight engines in Table 1, among them Poppler’s `-layout` and `-raw` modes, Marker 2.0.0 on PDFium and pdfplumber 0.11.10 on pdfminer, read all pages of originals and copies; Tesseract read 200-dpi renders. The reference is Poppler `-layout` text of the copy. *Lam–alef recall* is the share of reference words with lam + alef that an output reproduces; *word recall* ignores letter order.

Audit and retrieval. On 12 pages, AI-drawn region crops passed when their NFKC characters and digit groups matched, as multisets, the extracted words with page coordinates inside them; an AI reviewer then compared passing crops with the image in a blinded review (two passes, a third on disagreement). We reproduced exactly the BM25 and BGE-M3 dense retrieval of our earlier note [7]: 48 Arabic questions, 12 with their known passage in these PDFs, over 910 dictionary-repaired Arabic passages from these PDFs and 90 clean ones from web pages. We compared it with a copy in which every lam + alef in the PDF passages became alef + lam, and page indexes of originals and copies.

3 Results

Table 1: The lam–alef ligature by engine on 259 pages. Canary: alef + (alef, hamza-alef or madda-alef) + lam, a sequence no Arabic word contains, per 1,000 words. Recall against Poppler –layout text of the copy; pdfminer’s after reordering.

Engine	Canary, original	Canary, copy	Lam–alef recall	Word recall	Verdict
Poppler 26.07.0	67.4	0.02	0.04	1.00	reverses
MuPDF (PyMuPDF 1.28.2)	67.9	0.01	0.04	0.98	reverses
PDFium (pypdfium2 5.14.0)	67.6	0.07	0.04	0.98	reverses
pdf.js 6.4.299	68.0	0.01	0.03	0.98	reverses
pdfminer.six 20260107	0	0	0.04	0.99	reverses once reordered
PDFBox 3.0.8	0.02	67.6	0.99	0.98	correct
docling-parse (Docling 2.137.0)	0.01	68.4	0.93	0.92	correct
pypdf 6.20.0	0.06	69.3	0.97	0.96	unusable, glyph names
Tesseract 5.5.0, page images	0.06	—	0.95	0.94	correct

Four of eight engines reversed the ligature, a fifth once reordered; two kept it. All logical-order configurations of Poppler, MuPDF, PDFium and pdf.js produced 65 to 69 canary sites per 1,000 words and reproduced at most 4% of reference lam–alef words (Table 1). pdfminer writes display order, invisible to the canary; reversing its Arabic runs into reading order, as any consumer must, flipped its ligature too (4%). PDFBox reproduced 99% and docling-parse 93%, though Docling lost 8% of words; pypdf kept the order but leaked 9,937 glyph names such as /uni0627. In Poppler’s output خلال (*during*) appeared 223 times as خال and twice correctly (Figure 1a).

Why. All 86 lam–alef entries store lam before alef. A PDF stores Arabic glyphs in display order, left to right, so an extractor reverses each right-to-left run to recover reading order; reversing code point by code point also reverses the two characters of one glyph. The copy confirms this both ways: reversing engines fell to at most 0.07 canary sites per 1,000 words, pdfminer’s lam–alef recall rose to 0.99, and PDFBox and docling-parse rose to 68.

One word in eleven, with identical character counts. Against the copy, 11,234 of Poppler’s 126,504 words differ (8.9%; 8.2% to 10.1% per document), yet every page keeps its character multiset, so a check of characters and numbers passes all 259 pages. Other ligatures flip too: in one document في (*in*) came out as يف, among 62 such words.

Presentation-form pages are only partly spared. On 18 pages some fonts map lam–alef to one pre-shaped character, a presentation form, with no order to reverse; all 273 correct article forms in Poppler’s output occur there. Yet other fonts there gave 214 canary sites, and ligatures beyond the copy’s reach were reversed, such as مر written ري: about 21 two-letter swaps per 1,000 words against OCR; a blinded check confirmed 19 of 20.

A one-line test finds it in logical-order text. The canary regex [ا إ آ] ل matched on 204 of the 208 pages with at least 100 Arabic words in Poppler’s output; the other four have correct lam–alef, though three reverse other ligatures. It matched none of 15,940 words of web text, twice in the copy and no page of visual-order output. Per document, lam–alef words run 0.3 to 2.0 per 100 as extracted and 8.4 to 11.2 after the fix (Figure 1b).

A dictionary repair left a residue. Our corpus pipeline swapped letters at the article position, elsewhere only into known words. On PyMuPDF text it left 322 of 8,549 canary sites; 190 of 910 passages (20.9%) still have one.

OCR trades the reversal for lost words. Tesseract’s best model reproduced 95% of reference lam–alef words but missed or misread 4.8% of words, after excusing text-layer artefacts that three of five engines share. It added 3.5% spurious words and recovered 71% of number groups; the blinded check confirmed 20 of 24 substitutions and 16 of 16 dropped words. The vision-language parser PaddleOCR-VL-1.5 [8] never wrote the reversed form but missed 22.4% of words, added 22.9% spurious ones and recovered 52% of number groups, below our usability floor.

Top-four retrieval hits hid the damage; exact search did not. Of 55 lam–alef words in 31 of the 48 questions, 39 occur 2,127 times in repaired PDF passages, never reversed. For the 12 PDF-anchored questions the reversal left passage top-four hits unchanged (BM25 10, BGE-M3 8), but ranks moved: BM25 ranked the known passage lower for 1 question and higher for 3, BGE-M3 lower for 4 and higher for 1; page indexes gave 8 against 9 and 7 against 6 hits. An aggregate metric hid the damage; keyword filters, phrase search and quote checks fail on affected words.

Character-exact crops moved phrases and kept reversed ligatures. Of ten crops passing the multiset check, the AI review withheld two where a phrase had moved and four whole-column crops continuing on the next page. The four it kept held 41 canary sites and no correct article form; 48 of 316 words differ from the copy. On three two-column pages only geometry-based modes (Poppler –layout, pdfplumber in both modes, pypdf’s layout) mixed the columns, Poppler on 100 of 162 lines; Docling, pdfminer and PaddleOCR-VL read the left column first.

