# LongListBench: A Benchmark for Long-List Entity Extraction from Complex Business PDFs

Anton Fedoruk [1*] Serhii Shchoholiev [1†]
Akhil Mehta [1‡] Vishal Rohra [1§]

[1]Kay.ai, Brooklyn, NY, USA

July 2026

**Abstract**

Business PDFs can contain hundreds or thousands of target records, while most document benchmarks emphasize short forms or modest tables. LongListBench measures complete document-to-list extraction across 14 auditable layout, record-assembly, scale, and OCR stressors. It contains 32 synthetic PDFs, 29,599 target records, JSON ground truth, per-document stressor metadata, and OCR transcripts. Four repository-isolated coding-agent configurations are evaluated on the same transcripts. GPT-5.6-Sol and Claude Fable 5 recover 97.9% and 95.1% of records exactly, but reproduce only 8 and 9 of 32 complete document lists. Exact recall is 93.8% and 84.0% on structural challenges versus 99.5% for both on scale controls; secondary field-pair $F_1$ is 99.4% and 96.8%. Heterogeneous policy records separate the systems most sharply, at 73.3% and 14.6% exact recall. All visible content is synthetic; private documents informed structure only.

## 1 Introduction

Long-list entity extraction recovers hundreds or thousands of repeated records from semi-structured documents. It is common in insurance, finance, logistics, and procurement, but remains difficult to measure. A system may extract the first page correctly while dropping middle records, merging neighboring rows, or losing inherited context later in the document.

Established benchmarks such as FUNSD [1], SROIE [2], and CORD [3] emphasize key-value fields or short forms. DocVQA [4] and MP-DocVQA [5] test multi-page reasoning, but not exhaustive recovery of large repeated lists. DocILE [6] and VRDU [7] include

*anton@kay.ai
†serhii@kay.ai
‡akhil@kay.ai
§vishal@kay.ai

richer business documents, yet production extraction also encounters dense exports, repeated headers, OCR artifacts, split records, and fields distributed across distant packet sections.

LongListBench is designed to measure these failures. For each PDF, a system must reconstruct the entire target list under a declared output schema. The current release contains 32 single-PDF instances and 29,599 target records. It combines high-density scale controls with cross-section joins, long-range evidence, distractors, and heterogeneous policy records. All names, identifiers, values, and visible prose are synthetic; private documents were used only to study recurring layout and packet structure.

The release makes three contributions:

- A public corpus organized by 14 per-document stressor labels.
- Auditable artifacts for every instance: PDF, HTML source, OCR transcript, JSON ground truth, and per-document metadata.
- Strict record and document completeness metrics, secondary field diagnostics, report regeneration, and reference extraction protocols.

LongListBench is designed as a benchmark for measuring extraction systems under realistic long-list document conditions. Results should identify the input condition and extraction protocol and report strict completeness by evaluation role, document family, and stressor rather than relying on one partial-credit aggregate.

# 2 Related Work

Research on information extraction (IE) from visually rich documents has produced a broad ecosystem of datasets and models. However, much of the public evaluation landscape emphasizes either short documents (e.g., forms) or key-value extraction, leaving long-list entity extraction underexplored.

## 2.1 Document IE benchmarks

Early and widely used benchmarks such as FUNSD [1] focus on form understanding in noisy scans. Receipt datasets such as SROIE [2] and CORD [3] emphasize OCR, key fields, and relatively short item structures in narrow document types. These benchmarks are valuable, but typically contain relatively short documents and do not directly stress long lists of repeated entities.

DocILE [6] broadens the scope to business documents and includes line-item recognition, which is closer in spirit to long-list extraction. Document question answering benchmarks such as DocVQA [4] and MP-DocVQA [5] show that reasoning over document pages, especially multi-page documents, remains difficult; however, they evaluate question answering rather than recovery of full repeated record sets. VRDU [7] further argues that hierarchical and repeated fields (e.g., invoice line items) remain difficult for LLM-based extraction. More recent parsing-oriented benchmarks such as OmniDocBench [8] expand coverage across diverse PDF document types and evaluation tasks, but their primary emphasis is broad document parsing rather than repeated-entity extraction under long-list stressors.

Two concurrent industry benchmarks are especially close to our setting. LongArray-Extract [9] releases 45 synthetic PDFs with 29,328 target rows across clinical, financial, and legal documents. It isolates complete extraction of one long array and measures output-scale failures such as dropped or truncated rows. LongListBench provides per-document labels for 14 cross-cutting stressors in insurance and trucking documents.

LongExtractionBench [10], commissioned by Reducto and published by micro1, evaluates seven extraction systems on 225 long public documents and releases a 50-document inspection subset. It emphasizes completion rate, recall, failure modes, and latency across direct models and dedicated extraction platforms. LongListBench instead releases its full synthetic corpus, ground truth, OCR condition, stressor metadata, saved baseline predictions, and scorer for reproducible method development.

Our benchmark complements these efforts by making long-list failure conditions explicit, auditable in rendered pages, and available as per-document evaluation slices.

### 2.2 Document understanding models

Layout-aware pretraining approaches such as LayoutLM [11] and LayoutLMv3 [12] jointly model textual content, document structure, and visual signals, yielding strong performance on a range of document understanding tasks. In parallel, OCR-free approaches such as Donut [13] avoid explicit OCR by directly generating structured outputs from document images, mitigating OCR error propagation at the cost of specialized training. More recent layout-aware LLM and MLLM systems such as DocLLM [14] and DocLayLLM [15] push document understanding toward instruction-following, generative extraction, and richer multimodal reasoning.

In contrast, our work is model-agnostic: we provide rendered PDFs, OCR transcripts, and ground truth, enabling evaluation of OCR-based pipelines, OCR-free models, and LLM-based extraction. Our primary goal is to support reproducible measurement of long-list extraction robustness under realistic layout artifacts and long-range evidence.

## 3 Benchmark Construction

LongListBench[1] is a synthetic benchmark for long-list entity extraction in semi-structured business documents. Each instance consists of (i) structured JSON ground truth, (ii) rendered HTML and PDF artifacts, (iii) an OCR transcript in Markdown, and (iv) metadata describing target counts, artifact paths, and evidence patterns.

### 3.1 Entity schemas

The released corpus contains three broad target families.

**Claim incidents.** Claim-oriented documents use a loss-run incident schema with claim identifiers, policy metadata, claimant and driver fields, loss dates, coverage fields, status

[1]Code, data, and implementation details are available at https://github.com/kaydotai/longlistbench.

fields, and nested financial breakdowns. These records use strict normalized-record comparison plus field diagnostics, including date normalization, claimant-list sorting, and monetary rounding.

**Policy records.** Policy-oriented documents use heterogeneous records for locations, coverages, forms, endorsements, exclusions, rating rows, and premium items. A policy packet is the contract document issued by an insurer; it combines declarations, covered locations or classifications, coverage limits and deductibles, rating or premium schedules, required forms, and endorsements that modify the base policy. The released packets cover Businessowners Policy (BOP), Workers Compensation (WC), and Commercial General Liability (CGL).

**Operations records.** The core operations documents contain dense repeated lists from business operations workflows, including mileage rows, multi-section return rows, tax inquiry rows, driver records, vehicle schedule rows, and external loss-run rows. These records use generic normalized-record comparison rather than being forced into the claim incident schema.

## 3.2 Document design and complexity stressors

Private production PDFs were reviewed only to identify recurring structures such as portal exports, spreadsheet schedules, loss-run tables, declarations, forms, endorsements, and long packet organization. The released documents are production-inspired, not facsimiles of source packets.

Figure 1 summarizes construction. Human corrections to production extraction surfaced recurring failure patterns, which engineers translated into seeded generators, family-specific templates, and the stressor taxonomy catalogued below. Generators write target records first, render them through HTML templates to PDF, and transcribe page images with vision OCR. Every visible name, identifier, value, and prose passage is generated. The release fixes the resulting HTML, PDF, OCR, ground-truth, and metadata artifacts for reproducible evaluation. The private template tooling used for the current production-like layouts is intentionally not released, so we do not claim bit-for-bit public regeneration of every PDF.

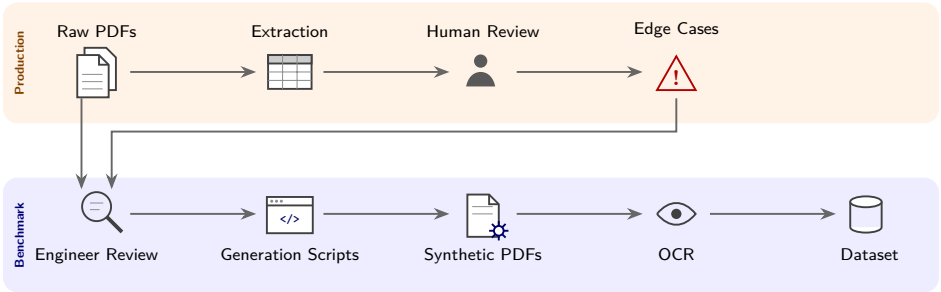



Figure 1: Dataset construction. Production failures inform structural templates; seeded generators write ground truth first, then render synthetic PDFs and OCR transcripts. No production values are released.

The released PDFs include dense tables, wrapped rows, form-like pages, and policy packet sections rather than only clean spreadsheet exports. The `stressors` metadata tracks 14 canonical cross-cutting stressors, catalogued below with representative rendered pages. The 45 distinct tokens currently found in `stressors` also include finer domain and implementation tags for audit slices; they are not additional canonical categories. We do not print benchmark labels inside the PDFs; that would make the documents less realistic and could make extraction easier. Instead, the stressors are visible in the layout and content organization, while the exact labels are recorded in sidecar metadata; Table 11 in the appendix maps every canonical stressor to representative pages for visual audit. Red boxes and labels in the figures below are annotations added for the paper; the released PDFs do not print them.

**`page_breaks`.** Target lists or supporting sections span page boundaries, often with repeated headers or inherited context. This is pagination pressure rather than an item-level split, and it is mostly a list-completeness stressor by itself. In Figure 2, a unit's mileage rows stop at a page boundary and resume on the next page under a repeated report header and a "Continued" unit band, so a per-page reader can drop the continuation or emit the repeated context as new records. The mileage exports apply this pressure at scale across dozens of pages.

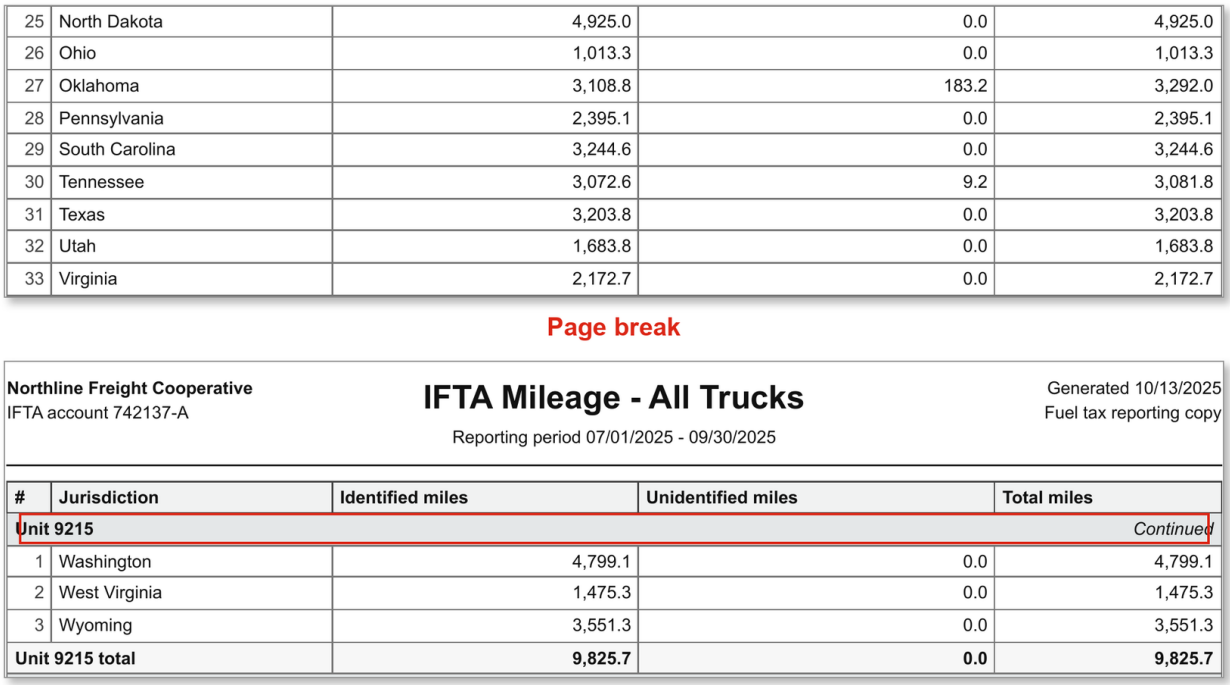
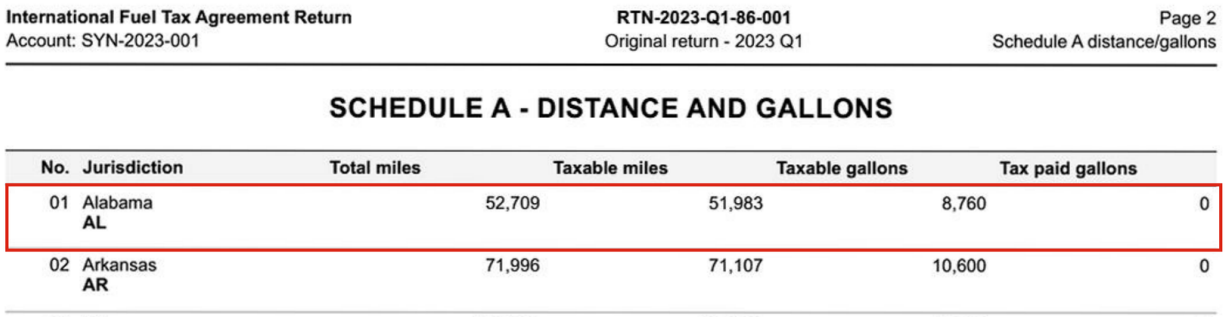



Figure 2: A unit's jurisdiction rows interrupted by a page break in a mileage export; the next page repeats the report header and marks the unit section as continued (boxed) before the rows and the unit total resume.

**`split_records`.** One logical target record has fields distributed across separate visual blocks, sections, or pages, so the extractor must assemble the item rather than copy one row. Figure 3 shows two row-level evidence blocks for one target in a multisection IFTA return. Return headers supply reporting context, Schedule A supplies mileage and gallons, and a later Jurisdictions table supplies tax fields. The sections order jurisdictions differently: Alabama is row 01 in Schedule A but row 09 in the Jurisdictions detail, so the extractor must join by return and jurisdiction rather than copy one row.



**International Fuel Tax Agreement Return** | **RTN-2023-Q1-86-001** | Page 2
Account: SYN-2023-001 | Original return - 2023 Q1 | Schedule A distance/gallons

**SCHEDULE A - DISTANCE AND GALLONS**

| No. | Jurisdiction | Total miles | Taxable miles | Taxable gallons | Tax paid gallons |
|---|---|---|---|---|---|
| 01 | Alabama **AL** | 52,709 | 51,983 | 8,760 | 0 |
| 02 | Arkansas **AR** | 71,996 | 71,107 | 10,600 | 0 |

(a) The Schedule A distance-and-gallons table of a multisection IFTA return.

**JURISDICTIONS**

| No. | D1 Jur | Jurisdiction Name | D2 Fuel | D3 Tax Rate | D8 Net Tax Gal. | D9 Tax Due/(Credit) | Surtax/Fees |
|---|---|---|---|---|---|---|---|
| 01 | OK | Oklahoma | DI | 0.3362 | 22,485 | $7,559.46 | $0.00 |
| 02 | PA | Pennsylvania | DI | 0.2729 | 11,286 | $3,079.95 | $116.82 |
| 03 | ID | Idaho | DI | 0.4850 | 6,578 | $3,190.33 | $0.00 |
| 04 | TN | Tennessee | DI | 0.5930 | 24,127 | $14,307.31 | $645.12 |
| 05 | WI | Wisconsin | DI | 0.5565 | 18,319 | $10,194.52 | $0.00 |
| 06 | FL | Florida | DI | 0.3693 | 2,512 | $927.68 | $0.00 |
| 07 | KY | Kentucky | DI | 0.2604 | 5,917 | $1,540.79 | $56.64 |
| 08 | NV | Nevada | DI | 0.5455 | -1,854 | ($1,011.36) | $0.00 |
| 09 | AL | Alabama | DI | 0.5213 | 8,760 | $4,566.59 | $0.00 |
| 10 | GA | Georgia | DI | 0.3277 | 4,972 | $1,629.32 | $0.00 |

(b) The Jurisdictions tax-detail table from a later section of the same packet.

Figure 3: One target record split across sections of the same packet.

**`cross_section_join` and `repeated_keys`.** `split_records` is the general condition: a record's fields do not sit in one contiguous block. The multisection returns in Figure 3

sharpen it twice. First, the pieces sit in separately labeled sections — the mileage and gallons in Schedule A, the tax fields in the Jurisdictions section, the reporting period in the return header — so the extractor must join all three to emit one row (`cross_section_join`). Second, the natural join key is ambiguous: the packet contains several returns and every return lists the same states, so "AL" alone matches many rows across the document, and joining on the state code mixes values from different returns; the correct key is the pair of return and jurisdiction (`repeated_keys`).

**`multi_row`.** One logical record includes wrapped notes, descriptions, clause prose, or continuation rows. In Figure 4, each incident's description, continuation notes, and ledger lines print below its main row. A one-row-one-record parser can fragment the target or attach text to the wrong item. The driver packets show the same pattern as multi-line MVR detail blocks.

| Effective Date | Claim Rptd | Claim Number | Claimant | Date Reported | Date of Loss | Status | Date Closed | Adjuster |
|---|---|---|---|---|---|---|---|---|
| 04/12/19 | X | 00928137-10000 | Canyon Fleet Body | 05/03/19 | 05/02/19 | Pending | | Morgan X. Larkin |
| Initial Claim Description: fuel-island scrape reported during yard maneuver; coverage was accepted subject to deductible application; salvage credit had not posted to the run<br>Continuation: payment history may exclude late-posted recovery activity. Unit TR-1000. | | | | | | | | |
| 04/12/19 | X | 00928137-10001 | Juniper Transit Yard | 05/15/19 | 05/09/19 | Pending | | Avery V. Gaines |
| Initial Claim Description: parking-lot contact reported by third-party claimant \| Driver: Jordan Y. Foster | | | | | | | | |
| 04/12/19 | X | 00928137-10002 | East Ridge Collision | 05/27/19 | 05/16/19 | Reopened | | Parker L. Ivers |
| Initial Claim Description: cargo load shifted during an abrupt traffic stop Unit TR-1034 / Operator Taylor L. Mercer | | | | | | | | |
| 04/12/19 | X | 00928137-10003 | Juniper Transit Yard | 06/08/19 | 05/23/19 | Closed | 08/20/19 | see ledger |
| Initial Claim Description: loading bay bollard contact reported by warehouse staff Unit TR-1051 / Operator Vaughn R. Ellis | | | | | | | | |
| Claim ledger: adjuster Reese R. Carver; reserve $0.00; paid $9,261.01; recovery $0.00. | | | | | | | | |

Figure 4: Multi-row record details in an external loss run: the boxed description and continuation lines belong to the claim row above them.

**`merged_cells`.** Some cells span several columns instead of respecting the one-cell-per-column grid, so a parser that assumes a regular grid misaligns everything after them. The merges in this release are horizontal (`colspan`): in Figure 5, a subtotal cell spans the leading columns of a mileage table and interrupts otherwise regular unit rows. The roll-up is context, not a target record, so the extractor must skip it without ending the table or emitting it as a data row. The loss-run tables use the same device for description and total cells that stretch under several columns. Vertical merges, where one cell applies to several rows, do not appear in the current corpus.

| | | | | | |
|---|---|---|---|---|---|
| 528 | Virginia | 2,500.7 | 2,500.7 | 0.0 | |
| 528 | Washington | 4,236.7 | 4,074.8 | 161.9 | R |
| 528 | West Virginia | 5,355.8 | 5,355.8 | 0.0 | |
| 528 | Wisconsin | 96.2 | 96.2 | 0.0 | |
| **528 subtotal** | | **117,533.8** | **116,681.8** | **852.0** | |
| 9852 | Alabama | 2,130.8 | 2,130.8 | 0.0 | |
| 9852 | Arizona | 2,326.7 | merged subtotal distractor | 0.0 | |
| 9852 | Arkansas | 5,426.5 | 5,426.5 | 0.0 | |
| 9852 | Connecticut | 5,495.7 | 5,495.7 | 0.0 | |

Figure 5: A merged subtotal row interrupts a mileage table.

**`multiple_tables`.** A single page mixes the target table with other table-shaped content, so the extractor must select the right table, not only the right rows. In Figure 6, one loss-run page holds the target incident table, a policy-period total row whose dollar cells sit in the same columns as claim financials, and a complete second policy section that is an empty no-claims table with its own header. The totals and the empty section must be consumed without emitting anything: the total is not a ninth claim, and the correct output for the empty section is exactly zero records.

| Effective Date | Claim Rptd | Claim Number | Claimant | Date Reported | Date of Loss | Status | Date Closed | Adjuster |
|---|---|---|---|---|---|---|---|---|
| 04/12/19 | X | 00928137-10000 | Canyon Fleet Body | 05/03/19 | 05/02/19 | Pending | | Morgan X. Larkin |
| Initial Claim Description: fuel-island scrape reported during yard maneuver; coverage was accepted subject to deductible application; salvage credit had not posted to the run | | | | | | | | |
| Continuation: payment history may exclude late-posted recovery activity. Unit TR-1000. | | | | | | | | |
| 04/12/19 | X | 00928137-10001 | Juniper Transit Yard | 05/15/19 | 05/09/19 | Pending | | Avery V. Gaines |
| Initial Claim Description: parking-lot contact reported by third-party claimant \| Driver: Jordan Y. Foster | | | | | | | | |
| 04/12/19 | X | 00928137-10002 | East Ridge Collision | 05/27/19 | 05/16/19 | Reopened | | Parker L. Ivers |
| Initial Claim Description: cargo load shifted during an abrupt traffic stop Unit TR-1034 / Operator Taylor L. Mercer | | | | | | | | |
| 04/12/19 | X | 00928137-10003 | Juniper Transit Yard | 06/08/19 | 05/23/19 | Closed | 08/20/19 | see ledger |
| Initial Claim Description: loading bay bollard contact reported by warehouse staff Unit TR-1051 / Operator Vaughn R. Ellis | | | | | | | | |
| Claim ledger: adjuster Reese R. Carver; reserve $0.00; paid $9,261.01; recovery $0.00. | | | | | | | | |
| 04/12/19 | X | 00928137-10004 | Ironwood Trailer Works | 06/20/19 | 05/30/19 | Pending | | Morgan Y. Osborne |
| Initial Claim Description: loading bay bollard contact reported by warehouse staff Driver not supplied in carrier export | | | | | | | | |
| 04/12/19 | X | 00928137-10005 | Palisade Container | 07/02/19 | 06/06/19 | Closed | 10/05/19 | Morgan G. Keller |
| Initial Claim Description: forklift contact with parked trailer during unloading Scheduled driver noted separately: Harper R. Gaines | | | | | | | | |
| Financial detail: reserve carried as $0.00; payments $14,935.44; recoveries $302.00. | | | | | | | | |
| 04/12/19 | X | 00928137-10006 | Ironwood Trailer Works | 07/14/19 | 06/13/19 | Pending | | Taylor O. Gaines |
| Initial Claim Description: minor bodily injury allegation after a lane-change contact \| Driver: Vaughn C. Benton | | | | | | | | |
| Section note: Closed claims can reopen if a supplemental medical bill or repair invoice is received. | | | | | | | | |
| 04/12/19 | X | 00928137-10007 | Alder Logistics Repair | 07/26/19 | 06/20/19 | Closed | 11/20/19 | see ledger |
| Initial Claim Description: fuel-island scrape reported during yard maneuver; rental and downtime amounts were still being reconciled \| Driver: Quinn K. Ellis | | | | | | | | |
| Claim ledger: adjuster Jordan Z. Delaney; reserve $0.00; paid $20,609.52; recovery $0.00. | | | | | | | | |
| Continuation: payment history may exclude late-posted recovery activity. Unit TR-1119. | | | | | | | | |
| **Policy period total for SC00928137** | | | | | | | | |

INSURED NAME **Northstar Cartage LLC** | CUSTOMER NO. **C043101** | POLICY NO. **SC00928138**

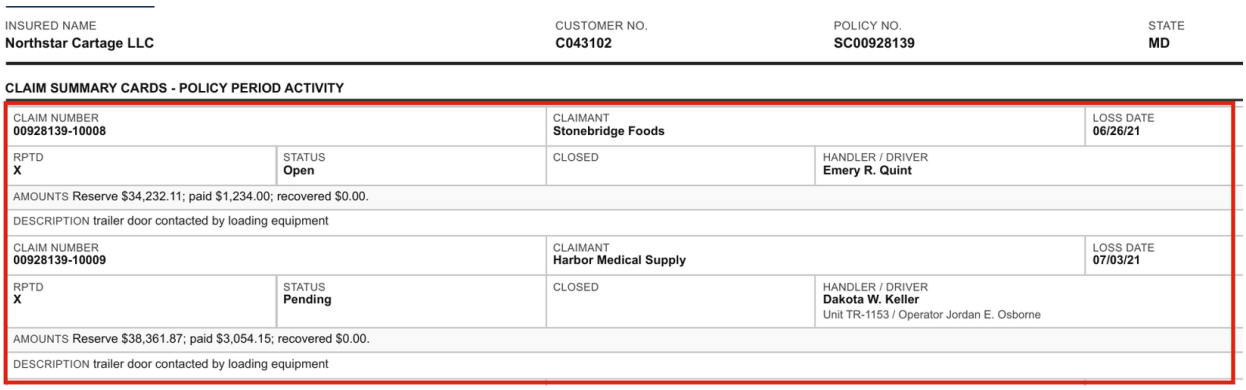
| Effective Date | Claim Rptd | Claim Number | Claimant | Date Reported | Date of Loss | Status | Date Closed | Adjuster |
|---|---|---|---|---|---|---|---|---|
| **No Claims Reported as of: 02/26/2026** | | | | | | | | |
| **Policy period total** | | | | | | | | |

Figure 6: One loss-run page with three table-shaped regions: the target incident table, a boxed policy-period total row, and a boxed empty no-claims section for a second policy. Only incident rows are targets.

**`large_doc`.** Hundreds to thousands of targets, or enough pages to expose truncation and list-completeness failures. This stressor is a scale property rather than a visible layout feature: the longest released files reach 133 and 148 pages, and single files reach four-digit target counts.

**`ocr_condition` and `ocr_layout_condition`.** The released text input is OCR output from rendered page images rather than the PDF text layer or the HTML source (`ocr_condition`), and the transcripts preserve visual spacing and reading order instead of converting tables into clean CSV-style rows (`ocr_layout_condition`). These are input conditions rather than layout features; Section 3.5 describes the OCR pipeline and how reported results should state the input condition.

**`duplicates`.** Duplicate handling cuts both ways. The output is scored as a multiset, so if a document legitimately contained two identical records, both copies would be required; a blanket "drop duplicates" rule is therefore not safe. The pressure in the current corpus comes from the opposite direction: near-duplicate *restatements* that must not be emitted at all. In Figure 7, summary cards restate claim rows whose detail tables appear later in the same report, so each claim is one target record, not two, and the policy packets add prior-term and archived sections with the same shape as current material. The current release contains no true duplicate target records, so it stresses only the ignore side; pairing both directions is future work (Section 6).



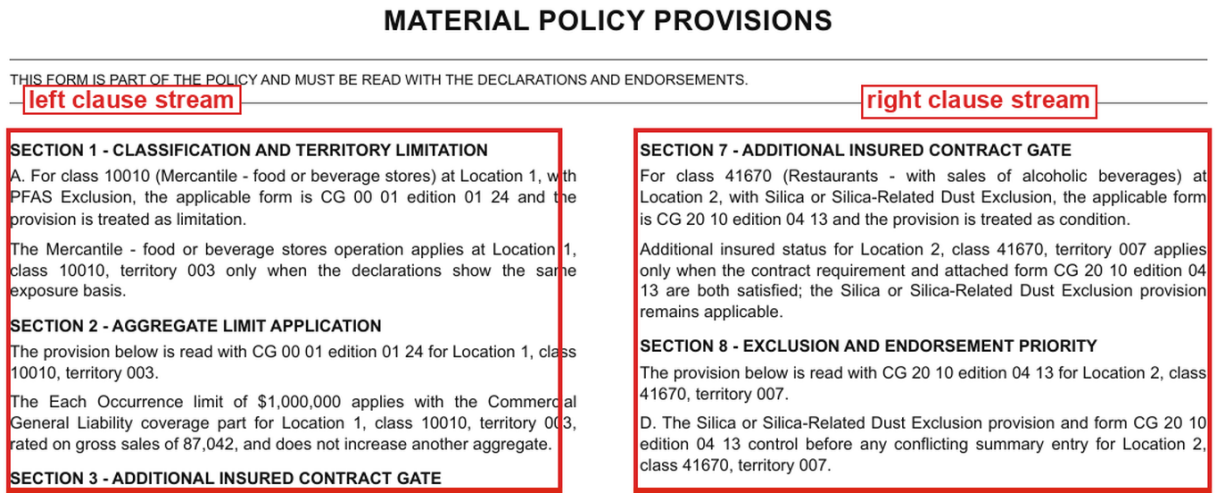
| INSURED NAME | CUSTOMER NO. | POLICY NO. | STATE |
|---|---|---|---|
| **Northstar Cartage LLC** | **C043102** | **SC00928139** | **MD** |

**CLAIM SUMMARY CARDS - POLICY PERIOD ACTIVITY**

| CLAIM NUMBER | | CLAIMANT | | LOSS DATE |
|---|---|---|---|---|
| **00928139-10008** | | **Stonebridge Foods** | | **06/26/21** |
| RPTD **X** | STATUS **Open** | CLOSED | HANDLER / DRIVER **Emery R. Quint** | |
| AMOUNTS Reserve $34,232.11; paid $1,234.00; recovered $0.00. | | | | |
| DESCRIPTION trailer door contacted by loading equipment | | | | |
| CLAIM NUMBER | | CLAIMANT | | LOSS DATE |
| **00928139-10009** | | **Harbor Medical Supply** | | **07/03/21** |
| RPTD **X** | STATUS **Pending** | CLOSED | HANDLER / DRIVER **Dakota W. Keller** Unit TR-1153 / Operator Jordan E. Osborne | |
| AMOUNTS Reserve $38,361.87; paid $3,054.15; recovered $0.00. | | | | |
| DESCRIPTION trailer door contacted by loading equipment | | | | |

Figure 7: Summary cards in a loss run restate claim rows whose detail tables appear later in the same report, creating near-duplicate distractors.

**`multi_column`.** Two-column and form-like layouts stress reading order. In Figure 8, a material-provisions page places two independent clause streams side by side; the clauses are target records in the policy packets, and naive left-to-right line order interleaves fields from unrelated clauses.



**MATERIAL POLICY PROVISIONS**

THIS FORM IS PART OF THE POLICY AND MUST BE READ WITH THE DECLARATIONS AND ENDORSEMENTS.

left clause stream

**SECTION 1 - CLASSIFICATION AND TERRITORY LIMITATION**

A. For class 10010 (Mercantile - food or beverage stores) at Location 1, with PFAS Exclusion, the applicable form is CG 00 01 edition 01 24 and the provision is treated as limitation.

The Mercantile - food or beverage stores operation applies at Location 1, class 10010, territory 003 only when the declarations show the same exposure basis.

**SECTION 2 - AGGREGATE LIMIT APPLICATION**

The provision below is read with CG 00 01 edition 01 24 for Location 1, class 10010, territory 003.

The Each Occurrence limit of $1,000,000 applies with the Commercial General Liability coverage part for Location 1, class 10010, territory 003, rated on gross sales of 87,042, and does not increase another aggregate.

**SECTION 3 - ADDITIONAL INSURED CONTRACT GATE**

right clause stream

**SECTION 7 - ADDITIONAL INSURED CONTRACT GATE**

For class 41670 (Restaurants - with sales of alcoholic beverages) at Location 2, with Silica or Silica-Related Dust Exclusion, the applicable form is CG 20 10 edition 04 13 and the provision is treated as condition.

Additional insured status for Location 2, class 41670, territory 007 applies only when the contract requirement and attached form CG 20 10 edition 04 13 are both satisfied; the Silica or Silica-Related Dust Exclusion provision remains applicable.

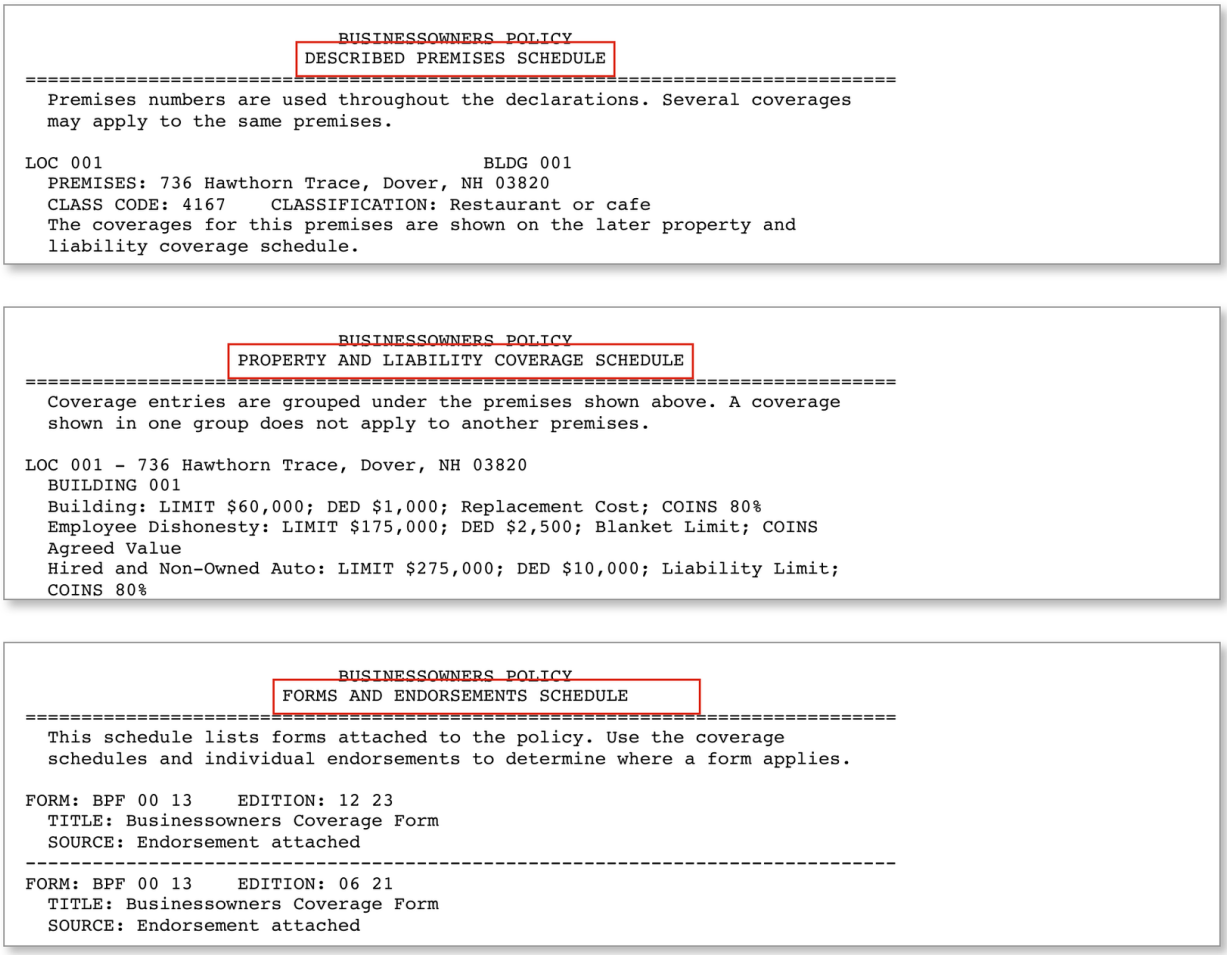
**SECTION 8 - EXCLUSION AND ENDORSEMENT PRIORITY**

The provision below is read with CG 20 10 edition 04 13 for Location 2, class 41670, territory 007.

D. The Silica or Silica-Related Dust Exclusion provision and form CG 20 10 edition 04 13 control before any conflicting summary entry for Location 2, class 41670, territory 007.

Figure 8: Two-column policy provisions with independent clause streams.

**heterogeneous_record_list.** A policy packet is scored as one output list, but its records are not one kind of row: a location record carries address, building, and class fields; a coverage record carries limits and deductibles; a form record carries form numbers and edition dates. Figure 9 shows the three schedules from one packet — premises, coverage, and forms — each printing records with a different shape. The stressor tests whether an extractor can emit records of several schemas in a single pass over one document. A system that locks onto one record layout for the whole document flattens or drops the families that do not fit it, which is visible in the results as a large gap on the policy packets.



```
                        BUSINESSOWNERS POLICY
                    DESCRIBED PREMISES SCHEDULE
================================================================================
  Premises numbers are used throughout the declarations. Several coverages
  may apply to the same premises.

LOC 001                                   BLDG 001
  PREMISES: 736 Hawthorn Trace, Dover, NH 03820
  CLASS CODE: 4167    CLASSIFICATION: Restaurant or cafe
  The coverages for this premises are shown on the later property and
  liability coverage schedule.
```

```
                        BUSINESSOWNERS POLICY
                PROPERTY AND LIABILITY COVERAGE SCHEDULE
================================================================================
  Coverage entries are grouped under the premises shown above. A coverage
  shown in one group does not apply to another premises.

LOC 001 - 736 Hawthorn Trace, Dover, NH 03820
  BUILDING 001
  Building: LIMIT $60,000; DED $1,000; Replacement Cost; COINS 80%
  Employee Dishonesty: LIMIT $175,000; DED $2,500; Blanket Limit; COINS
  Agreed Value
  Hired and Non-Owned Auto: LIMIT $275,000; DED $10,000; Liability Limit;
  COINS 80%
```

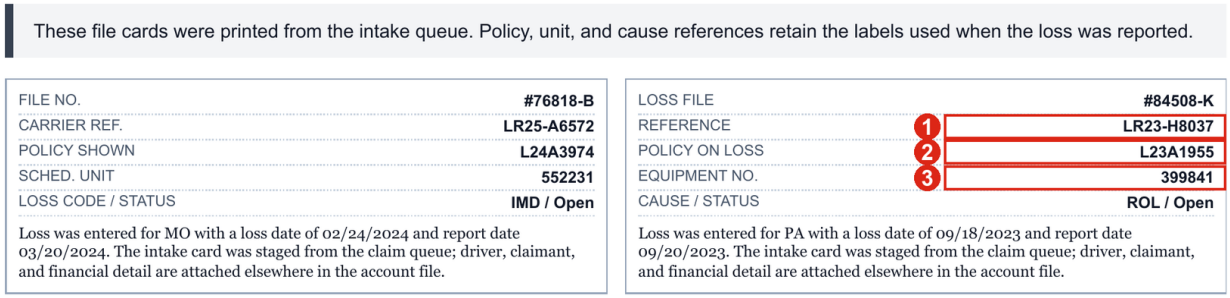
```
                        BUSINESSOWNERS POLICY
                    FORMS AND ENDORSEMENTS SCHEDULE
================================================================================
  This schedule lists forms attached to the policy. Use the coverage
  schedules and individual endorsements to determine where a form applies.

FORM: BPF 00 13    EDITION: 12 23
  TITLE: Businessowners Coverage Form
  SOURCE: Endorsement attached
--------------------------------------------------------------------------------
FORM: BPF 00 13    EDITION: 06 21
  TITLE: Businessowners Coverage Form
  SOURCE: Endorsement attached
```

Figure 9: Three schedules of one policy packet (titles boxed): location records, coverage records, and form records follow different schemas but land in the same output list.

The current release prioritizes fewer, larger, more realistic PDFs rather than many small template variants. This choice reflects the intended use case: measuring list-completeness failures at scale. Several PDFs contain more than 1,000 target records, and the full corpus contains 29,599 target records across 32 PDFs.

**`long_range_evidence.`** Required fields are separated by distant pages within the same PDF. Figure 10 illustrates long-range evidence for one claim. The target begins on page 3; three join keys recur in distant sections. These crops show only part of the full evidence chain.



These file cards were printed from the intake queue. Policy, unit, and cause references retain the labels used when the loss was reported.

| | |
|---|---|
| FILE NO. | #76818-B |
| CARRIER REF. | LR25-A6572 |
| POLICY SHOWN | L24A3974 |
| SCHED. UNIT | 552231 |
| LOSS CODE / STATUS | IMD / Open |

Loss was entered for MO with a loss date of 02/24/2024 and report date 03/20/2024. The intake card was staged from the claim queue; driver, claimant, and financial detail are attached elsewhere in the account file.

| | |
|---|---|
| LOSS FILE | #84508-K |
| ① REFERENCE | LR23-H8037 |
| ② POLICY ON LOSS | L23A1955 |
| ③ EQUIPMENT NO. | 399841 |
| CAUSE / STATUS | ROL / Open |

Loss was entered for PA with a loss date of 09/18/2023 and report date 09/20/2023. The intake card was staged from the claim queue; driver, claimant, and financial detail are attached elsewhere in the account file.

(a) Page 3: a claim card carries reference `LR23-H8037` ①, policy `L23A1955` ②, and unit `399841` ③.

| | |
|---|---|
| ACCOUNT POLICY | L23A1604 |
| INSURED ACCOUNT | Port James Cargo |
| OPERATING COMPANY | Port James Cargo |
| OPERATING DIVISION | General - OH |

Producer or agency note: Brokerage 008. Runoff cards are retained for account history only.

| | |
|---|---|
| ② POLICY REF. | L23A1955 |
| NAMED INSURED | Rollinschester Hauling |
| RISK NAME | Rollinschester Hauling |
| STATE/DIVISION | General - OH |

Producer or agency note: Brokerage 002. Runoff cards are retained for account history only.

(b) Page 28: the policy register resolves policy ② to the named insured.

| | |
|---|---|
| SCHEDULED AUTO | 2023 PB 399841 |
| ③ ASSIGNED DRIVER | Ramsey, Taylor |
| OPERATING ACCOUNT | Rollinschester Hauling - OH |

Roster status: active scheduled unit. Dispatch reference on file: LR23-H8037.

| | |
|---|---|
| ① ACCOUNTING REF | LR23-H8037 |

Accounting close extract; match this reference to the intake card before assigning claim number.

| | | | | |
|---|---|---|---|---|
| **BI** | Reserve **$0.00** | Paid **$0.00** | Recovered **$0.00** | Incurred **$0.00** |
| **PD** | Reserve **$36,010.10** | Paid **$0.00** | Recovered **$0.00** | Incurred **$36,010.10** |
| **LAE** | Reserve **$0.00** | Paid **$0.00** | Recovered **$0.00** | Incurred **$0.00** |
| **DED** | Reserve **$0.00** | Paid **$0.00** | Recovered **$0.00** | Incurred **$0.00** |

(c) Page 26 matches unit ③ to a driver; page 60 matches reference ① to financial fields.

Figure 10: Distant references for one target claim. The complete record uses six evidence sections; unrelated rows are cropped out.

The crops are illustrative, not complete provenance. The scored claim has 24 top-level fields plus nested financials. Its metadata maps six evidence blocks: intake cards, driver assignments, policy registers, cause narratives, claimant notices, and financial ledgers. The first and last primary blocks are 57 pages apart, so extraction must preserve several join keys across the packet.

## 3.3 Corpus inventory

The release favors fewer, larger PDFs: 26 operations documents contain 28,178 targets, three claim packets contain 77, and three policy packets contain 1,344. Table 10 in the appendix gives the complete technical family inventory.

## 3.4 Evaluation configurations

Each HF instance belongs to one of three release configurations (Table 1). The public column `complexity_regime` identifies its document family, while `stressors` lists the cross-cutting conditions catalogued in Section 3.2. In the source manifest, configuration membership is `hf_config` and the stressor list is `problems`; export renames the latter to `stressors`. A PDF can therefore be large, multi-table, OCR-conditioned, and long-range at the same time.

These configurations are not a difficulty ladder. Core operations includes both parser-friendly scale controls and multisection joins. Claim and policy packets emphasize inherited context, distant evidence, heterogeneous schemas, and distractors. Results should therefore be sliced by evaluation role, family, and stressor rather than reduced to one ranking.

Table 1: Evaluation configurations in the current release.

| Configuration | PDFs | Records | Main contents |
|---|---|---|---|
| Core operations | 26 | 28,178 | High-density repeated records, production-like tables and exports, OCR preservation, cross-section joins, and output completeness. |
| Claim multi-hop | 3 | 77 | Long-range joins from claim schedules to policy registers, driver rosters, claimant indexes, cause-code appendices, and ledgers. |
| Policy packets | 3 | 1,344 | Heterogeneous policy records across declarations, locations or classifications, forms, material clauses, endorsements, rating schedules, and premium summaries. |
| **Total** | **32** | **29,599** | |

### 3.5 OCR condition

Every released PDF has an OCR transcript generated from rendered page images. The released transcript condition was produced with Google Gemini 3.5 Flash vision OCR through the direct Google/Vertex Gemini API path, after rasterizing PDF pages at 200 DPI. The OCR transcript is the released text condition for this version; it is not the PDF text layer or the HTML source text. We treat OCR as a controlled input condition: a system may evaluate directly from the PDF, from the OCR transcript, or from its own OCR pipeline, but reported results should state the input condition clearly.

### 3.6 Long-range evidence

LongListBench includes single-document multi-hop cases because many production tasks are not contained in one adjacent table region. A target loss-run record may require joining a front claim schedule against policy registers, driver rosters, cause-code appendices, claimant indexes, and payment ledgers that appear dozens or hundreds of pages later. Policy packets require similar joins across declarations, location schedules, forms, endorsements, rating pages, and premium summaries.

This design is intended to stress extraction systems that can inspect long documents, preserve join keys across distant sections, and verify record completeness. It is also intended to make future agentic and retrieval-loop methods measurable without changing the public data format: each sample remains one PDF with one ground-truth list.

## 4 Evaluation

For each PDF, the system must reconstruct the complete target list under a declared output contract. Predictions use `incidents` for claim tasks and `records` for operations, external loss-run, and policy tasks.

### 4.1 Input condition

Every PDF has a released OCR transcript. Experiments may instead use direct PDF input or another OCR system, but must report the input condition, model, protocol, and included samples.

### 4.2 Extraction protocols

The repository includes a raw full-context API adapter, an OpenAI Agents SDK sandbox adapter, and repository-denied Codex and Claude Code CLI runners. Four full-corpus CLI results use the same OCR-conditioned protocol.

**Full-context one-shot extraction.** The entire transcript is sent in one request. This is a useful lower-bound calibration, but long outputs may truncate or time out.

**Agentic sandbox extraction.** The runner creates a temporary workspace containing the transcript, field contract, prompt, and output directory. A macOS sandbox profile denies the benchmark repository, while the task prompt prohibits using files outside the workspace. This is repository isolation, not a host-wide filesystem allowlist. The model may search, parse, and verify the current document before writing JSON.

Claim runs receive the published JSON Schema. Generic-record runs receive sample-specific field names and record groups derived from the ground-truth object structure before workspace creation. This exposes the required output schema, not target values or counts. Ground-truth files, generator code, target values, and target counts are not copied into the workspace, and the repository path is denied.

**Parser diagnostics.** Template parsers are useful for structured scale-control families. A frozen parser evaluated on unseen templates measures transfer robustness, while a per-document agent may write code for the current input. These protocols answer different questions and must be labeled separately.

**Why use an agentic baseline.** The largest targets contain thousands of rows, so a single response can hit output or latency limits before returning a complete list. Fixed chunking avoids that limit but introduces chunk size, overlap, boundary repair, and merge parameters that can dominate the result. A per-document sandbox instead lets the extractor choose how to inspect, segment, parse, and verify each new document under one output contract. We use this as a reproducible full-corpus reference protocol, not as a raw measure of either base model alone.

## 4.3 Configuration-aware reporting

Field-pair micro-$F_1$ can hide incomplete records because it gives partial credit and large scale-control families dominate field counts. Reports should lead with exact-record recall and document completion. Secondary reporting includes field diagnostics, predicted counts, configuration and family results, and stressor slices such as `cross_section_join`, `long_range_evidence`, and `heterogeneous_record_list`. We classify four parser-friendly families as scale controls and the remaining seven families as structural challenges using a fixed family mapping.

## 4.4 Validation and report regeneration

Reports are stored as JSON and Markdown. The checker recomputes every metric from saved predictions and ground truth without rerunning a model.

## 4.5 Strict completeness and field diagnostics

The primary metrics are exact-record recall and complete-document success. A predicted record is exact only when every normalized target field equals one ground-truth record. We

compute

$$\text{ExactRecordRecall} = \frac{|R_G \cap R_P|}{|R_G|}, \tag{1}$$

$$\text{CompleteDocument} = \mathbb{1}[R_G = R_P], \tag{2}$$

where $R_G$ and $R_P$ are the normalized ground-truth and predicted record multisets. The intersection preserves duplicate multiplicity, and $\mathbb{1}[R_G = R_P]$ is one only when neither records nor duplicates are missing or extra. Source order is not part of the released contract.

For claim incidents, records are keyed by normalized incident identifiers. String values are whitespace-normalized and compared case-insensitively, dates are normalized to `MM/DD/YYYY`, claimant lists are sorted, zero-valued default financial-breakdown cells are omitted from field diagnostics, and non-zero monetary fields are rounded to cents. Generic records use the same string normalization. Unambiguous labels are canonicalized: jurisdiction names to codes, fuel descriptions to parenthetical codes, line-of-business names to acronyms, visible `Applies within` wrappers to their core clause scope, visible `Unit` prefixes in vehicle identifiers, and `Quarter Return`/`Quarterly Return` heading aliases. Extra heading context remains an error. For field diagnostics, exact generic records are anchored first and remaining records are matched deterministically by overlapping non-global field pairs. `record_type`, global policy fields, and hidden row identifiers are excluded from field matching and field-pair scoring. These field-level exclusions do not relax exact-record comparison, which uses the full normalized public record.

For secondary partial-credit diagnostics we compute

$$\text{found} = |G \cap P|, \tag{3}$$

$$\text{recall} = \frac{\text{found}}{|G|}, \tag{4}$$

$$\text{precision} = \frac{\text{found}}{|P|}, \tag{5}$$

$$\text{F1} = \frac{2\,\text{precision}\,\text{recall}}{\text{precision} + \text{recall}}, \tag{6}$$

where $G$ and $P$ are the normalized ground-truth and predicted field-value multisets. Reports also include exact-record precision/$F_1$, predicted counts, missing and extra records, document-macro field $F_1$, and family, evaluation-role, and stressor summaries. Because stressors co-occur, stressor slices describe associations rather than causal effects.

# 5 Results

This revision reports corpus validation and four OCR-conditioned agentic extraction baselines on the released corpus of 32 documents. Earlier prerelease experiments used a development set of 80 documents with different templates, schemas, and scoring. We retain those experiments as development history but do not mix their scores with the current release.

## 5.1 Artifact completeness

Each manifest instance has one PDF, HTML source, JSON ground-truth file, metadata file, and OCR transcript. The release contains 28,255 non-policy records and 1,344 policy records, for 29,599 targets overall. Table 2 summarizes the release gates.

Table 2: Current release validation summary.

| Check | Value |
|---|---|
| Manifest instances | 32 |
| PDF, HTML, ground-truth, metadata, and OCR artifacts | 32 each |
| Total target records | 29,599 |
| Released transcript condition | OCR |
| Average identifier coverage | 100.0% |
| Tracked identifier-field support | 100.0% |
| Records with missing tracked OCR identifiers | 0 |
| Audited numeric OCR misses | 56 of 76,968 (0.073%) |

The largest scale family, mileage by vehicle, contributes 17,565 records across eight PDFs. Structural families add sparse enrichment, split schedules, mixed row/detail blocks, heterogeneous records, and evidence distributed across sections. All 32 OCR transcripts pass the 95% identifier-coverage gate. A numeric audit records 56 genuine OCR misses among 76,968 checked ground-truth numeric fields with absolute value at least 10. The transcripts are not hand-corrected, and the exact miss set is released. These checks bound the OCR condition; they are not extraction scores.

## 5.2 Agentic OCR baselines

Each run received one OCR transcript and the public field contract in a temporary workspace. A macOS sandbox denied the benchmark repository, and ground truth, target values and counts, and generator code were absent. Agents could inspect the transcript, write temporary code, validate JSON, and return the complete list. These are protocol baselines, not raw one-shot model measurements.

The initial full-corpus runs were produced on July 14, 2026 at xhigh effort using Codex CLI 0.144.4 or Claude Code 2.1.209. After correcting the driver/MVR sparse-enrichment contract, those three inputs were rerun on July 21 with Codex CLI 0.144.6 or Claude Code 2.1.216; the other 29 unchanged predictions were replayed. Claude Code safe mode disabled project instructions, skills, plugins, hooks, and MCP servers. Per-sample metadata records model routing, input fingerprints, and prediction hashes. All reports cover the same 29,599-target manifest with zero execution errors, and saved predictions reproduce every score in Table 3.

Table 3: Full-corpus agentic OCR baselines. Strict completeness is primary; field $F_1$ is secondary partial credit.

| Agent and model | Exact recall | Complete docs | Micro-$F_1$ | Macro-$F_1$ |
|---|---|---|---|---|
| Codex CLI, GPT-5.6-Sol (xhigh) | 97.9% | 8/32 (25.0%) | 99.4% | 99.4% |
| Claude Code, Fable 5 (xhigh) | 95.1% | 9/32 (28.1%) | 96.8% | 93.6% |
| Codex CLI, GPT-5.5 (xhigh) | 94.5% | 4/32 (12.5%) | 98.8% | 98.6% |
| Claude Code, Opus 4.8 (xhigh) | 97.7% | 7/32 (21.9%) | 99.4% | 99.3% |

No configuration completes more than 9 of 32 documents. GPT-5.6-Sol has the highest exact-record recall, Fable 5 completes the most documents, and Opus 4.8 has the highest field micro-$F_1$ by a narrow margin. The family results below are more informative than this narrow aggregate ranking.

We preassign four parser-friendly families as scale controls and the other seven as structural challenges. Table 4 reports the latest models by role.

Table 4: Strict completeness by evaluation role for the latest models.

| Role | Docs | Records | Sol exact | Fable exact | Sol complete | Fable complete |
|---|---|---|---|---|---|---|
| Structural challenges | 19 | 8,414 | 93.8% | 84.0% | 4/19 (21.1%) | 5/19 (26.3%) |
| Scale controls | 13 | 21,185 | 99.5% | 99.5% | 4/13 (30.8%) | 4/13 (30.8%) |

Table 5 uses problem-first labels so the extraction challenge is visible without domain knowledge. Labels map one-to-one to the public `complexity_regime` values.

Table 5: Latest-model exact-record recall by extraction problem. The divider separates structural challenges from scale controls.

| Extraction problem | Docs | Records | GPT-5.6-Sol | Fable 5 |
|---|---|---|---|---|
| Sparse record enrichment (driver/MVR) | 3 | 1,260 | 99.4% | 100.0% |
| Long-range claim joins | 3 | 77 | 98.7% | 98.7% |
| Split return schedules | 3 | 2,737 | 95.5% | 95.5% |
| Mixed row/detail loss runs | 3 | 900 | 97.3% | 92.2% |
| Tax inquiry detail tables | 2 | 1,300 | 99.9% | 99.8% |
| Heterogeneous policy records | 3 | 1,344 | 73.3% | 14.6% |
| Cross-section return joins | 2 | 796 | 99.6% | 99.6% |
| Tax-summary scale controls | 2 | 1,520 | 99.5% | 99.5% |
| Driver-schedule scale control | 1 | 500 | 99.8% | 99.8% |
| Mileage-by-vehicle scale controls | 8 | 17,565 | 99.4% | 99.4% |
| Vehicle-schedule scale controls | 2 | 1,600 | 100.0% | 100.0% |

The failures differ in kind. GPT-5.6-Sol matches 1,252 of 1,260 driver records and completes two packets; on the third it emits eight report summaries as separate partial rows instead of merging them into the roster records. Fable 5 joins every sparse report correctly and completes all three packets. Both systems match 2,614 of 2,737 split-schedule records and 76 of 77 long-range claim records. Policy packets separate them: GPT-5.6-Sol matches 985 of 1,344 records with 95.8% field $F_1$, while Fable 5 returns only 232 records, matches 196, and reaches 33.5% field $F_1$. Scale controls remain near ceiling, yet only four of 13 are completely correct for either system because a single missing, extra, or malformed record fails document completion.

Table 6 reports exact-record recall for all 14 canonical stressors. The largest observed gap occurs in heterogeneous policy lists. The long-range and multi-column rows cover the same six packets and are therefore not independent findings.

Table 6: Exact-record recall over documents carrying each canonical stressor tag. Tags are document-level and overlap, so these are observational slices rather than isolated effects.

| Stressor | Docs | Records | GPT-5.6-Sol | Fable 5 |
|---|---|---|---|---|
| *Record assembly and schema* | | | | |
| Split records | 15 | 9,797 | 94.9% | 86.9% |
| Cross-section joins | 2 | 796 | 99.6% | 99.6% |
| Repeated keys | 2 | 796 | 99.6% | 99.6% |
| Long-range evidence | 6 | 1,421 | 74.7% | 19.1% |
| Heterogeneous record lists | 3 | 1,344 | 73.3% | 14.6% |
| *Layout and distractors* | | | | |
| Page-spanning content | 32 | 29,599 | 97.9% | 95.1% |
| Multi-row records | 22 | 21,942 | 97.7% | 94.0% |
| Merged cells (horizontal spans only) | 4 | 3,043 | 96.7% | 95.2% |
| Multiple tables | 31 | 29,255 | 98.1% | 96.0% |
| Multi-column pages | 6 | 1,421 | 74.7% | 19.1% |
| Near-duplicate distractors (no true target duplicates) | 7 | 1,321 | 93.1% | 73.4% |
| *Scale and OCR input* | | | | |
| Large documents | 29 | 28,339 | 97.8% | 94.9% |
| OCR input | 32 | 29,599 | 97.9% | 95.1% |
| OCR-preserved multi-section layout | 2 | 796 | 99.6% | 99.6% |

Each row scores all targets in a tagged PDF, not only records printed on pages where the condition is visible. The multi-column slice, for example, contains 1,344 policy records and 77 claim records. Fable 5 matches 196 policy records and 76 claim records, so its 19.1% aggregate reflects broad policy-list omissions rather than an isolated reading-order failure. Page-spanning content and OCR input cover all 32 documents; cross-section joins, repeated keys, and OCR-preserved multi-section layout cover the same two return packets.

Full-context one-shot extraction is not a practical full-corpus protocol here. Small claim packets can return scoreable outputs, but larger outputs reach model limits or time out. We therefore treat one-shot prompting as a lower-bound operational stress test and use the per-document agentic protocol for full-corpus comparison.

# 6 Limitations and Intended Use

LongListBench is a measurement tool for long-list extraction under controlled synthetic conditions. Table 7 summarizes the current scope.

The documents are synthetic. This is necessary for public release, but it means the benchmark cannot replace evaluation on a private production corpus. We used private documents only as structural references for layout and packet organization; the released names, identifiers, values, and prose are generated. Some sections are intentionally composed to cover benchmark stressors, so the corpus should be read as production-inspired measurement data rather than exact replicas of source documents.

Table 7: Scope of the current LongListBench release.

| Covered | Not covered / out of scope |
|---|---|
| High-density repeated lists, cross-section joins, long-range evidence packets, and mixed-schema output lists | Non-English documents, handwriting, medical bills, invoices, purchase orders, and many other document families |
| Production-like synthetic layouts based on observed document structures | Full visual diversity of private production corpora and carrier-specific edge cases |
| OCR transcripts for every released PDF, including a limited scanned-image IFTA condition | Multiple OCR engines, broad scan-quality sweeps, bounding boxes, and reading-order supervision |
| Strict normalized-record completeness plus field diagnostics | Source-order scoring, semantic equivalence, downstream task metrics, and human adjudication of ambiguous fields |
| Agentic sandbox support and four full-corpus OCR baselines | Latency/cost-normalized comparisons across broader agentic and retrieval-loop protocols |

The current release includes one OCR transcript condition. This supports reproducible OCR-conditioned evaluation, but it does not isolate OCR penalty by itself. Measuring OCR penalty requires adding another transcript or direct-PDF condition and running the same extraction protocol across both.

The released output contract treats records as an unordered multiset. Exact-record and complete-document metrics therefore test values and multiplicity, not source order. An order-sensitive task should declare ordering in the contract and rerun extractors under that requirement rather than retroactively penalizing the saved predictions.

The coding-agent runs denied access to the benchmark repository but did not use a host-wide filesystem allowlist. Ground truth remained inside the denied repository. Future comparisons should use an allowlisted container or virtual machine and retain tool traces; the Claude result format did not include them.

OCR support should also be interpreted at the field level. A unique-identifier coverage check can look strong while a missing section header affects many records that inherit that header. The current validation reports unique identifier coverage, affected records for tracked identifiers, and a numeric-fidelity gate over ground-truth numeric values with absolute value at least 10. Richer future releases should extend this with row-local OCR checks and multiple OCR engines.

Finally, the corpus now contains several operations record types whose public schema is represented by the ground-truth examples and generic scorer rather than separate hand-written JSON Schema files for every family. Adding explicit schemas for each operations family would improve reproducibility for prompt-based and agentic extractors.

LongListBench is intended for configuration-, family-, and stressor-aware measurement of record completeness, field alignment, OCR-conditioned extraction, and long-range evi-

dence handling. Parser-friendly operations families are scale controls; claims about difficult extraction should rely on packet and cross-section families as well as aggregate scores.

Complexity slices are observational. Tags overlap, and document families differ in length, schema, and layout, so lower performance in a tagged slice does not identify one stressor as the cause. Paired renderings that hold records constant while adding one stressor would be required for causal attribution.

Two stressors are currently one-sided. Duplicate pressure comes only from near-duplicate restatements that must be ignored; the corpus contains no true duplicate target records, so a system cannot be tested on keeping legitimate repeats while dropping restatements, and a blanket drop-duplicates rule is never punished. Merged-cell pressure comes only from horizontal `colspan` structures; vertical merges, where one cell applies to several rows, do not appear. Future releases should pair true-duplicate targets with restatement distractors in the same documents and add vertically merged schedules.

The release is not a complete model leaderboard. It includes GPT-5.5, GPT-5.6-Sol, Opus 4.8, and Fable 5 under one agentic protocol, but no latency- or cost-normalized comparison across broader models and protocols. Full-context one-shot runs are impractical on the largest outputs, and synthetic public documents cannot replace private production evaluation.

# 7 Conclusion

LongListBench measures complete long-list extraction from complex business PDFs. Its 32 synthetic documents contain 29,599 targets and 14 canonical stressors spanning layout, record assembly, distractors, scale, and OCR input conditions. Every instance includes PDF, HTML, OCR, ground truth, and metadata, with recomputable strict-completeness and field-diagnostic scoring.

Across four agentic OCR baselines, GPT-5.6-Sol and Fable 5 recover 97.9% and 95.1% of target records exactly but complete only 8 and 9 of 32 documents. Exact-record recall falls to 93.8% and 84.0% on structural challenges while both reach 99.5% on scale controls. Near-perfect field overlap therefore does not imply complete extraction, and tagged complexities do not affect every system equally. Future comparisons should report strict completeness, input condition, extraction protocol, problem and stressor slices, and saved predictions alongside partial-credit field scores.

# Author Contributions

**Anton Fedoruk:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing (original draft), Writing (review & editing).
**Serhii Shchoholiev:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing (original draft), Writing (review & editing).
**Akhil Mehta:** Writing (review & editing).
**Vishal Rohra:** Project administration, Resources, Supervision, Writing (review & editing).

# References


[1] G. Jaume, H. K. Ekenel, and J.-P. Thiran, *Funsd: A dataset for form understanding in noisy scanned documents*, 2019. arXiv: 1905.13538 [cs.IR]. [Online]. Available: https://arxiv.org/abs/1905.13538

[2] Z. Huang et al., *Icdar2019 competition on scanned receipt ocr and information extraction*, 2021. DOI: 10.1109/ICDAR.2019.00244 arXiv: 2103.10213 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2103.10213

[3] S. Park et al., *Cord: A consolidated receipt dataset for post-ocr parsing*, Document Intelligence Workshop at NeurIPS, 2019. [Online]. Available: https://openreview.net/forum?id=SJl3z659UH

[4] M. Mathew, D. Karatzas, and C. V. Jawahar, "Docvqa: A dataset for vqa on document images," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2200–2209. [Online]. Available: https://arxiv.org/abs/2007.00398

[5] R. Tito, D. Karatzas, and E. Valveny, *Hierarchical multimodal transformers for multipage docvqa*, 2022. arXiv: 2212.05935 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2212.05935

[6] Š. Šimsa et al., *Docile benchmark for document information localization and extraction*, 2023. arXiv: 2302.05658 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2302.05658

[7] Z. Wang, Y. Zhou, W. Wei, C.-Y. Lee, and S. Tata, "Vrdu: A benchmark for visually-rich document understanding," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, arXiv:2211.15421, 2023. DOI: 10.1145/3580305.3599929 [Online]. Available: https://arxiv.org/abs/2211.15421

[8] L. Ouyang et al., *Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations*, 2024. arXiv: 2412.07626 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2412.07626

[9] J. Reyhan, J. Bajor, and C. Hao. "Longarray-extract: A benchmark for complete structured extraction at scale," Accessed: Jul. 12, 2026. [Online]. Available: https://www.extend.ai/resources/long-array-extraction-benchmark

[10] micro1 and Reducto. "Longextractionbench," Accessed: Jul. 12, 2026. [Online]. Available: https://www.micro1.ai/benchmark/long-extraction

[11] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, *Layoutlm: Pre-training of text and layout for document image understanding*, 2020. DOI: 10.1145/3394486.3403172 arXiv: 1912.13318 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1912.13318

[12] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, *Layoutlmv3: Pre-training for document ai with unified text and image masking*, 2022. arXiv: 2204.08387 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2204.08387

[13] G. Kim et al., *Ocr-free document understanding transformer*, 2022. arXiv: 2111.15664 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2111.15664

[14] D. Wang et al., *Docllm: A layout-aware generative language model for multimodal document understanding*, 2024. arXiv: 2401.00908 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2401.00908

[15] W. Liao, J. Wang, H. Li, C. Wang, J. Huang, and L. Jin, *Doclayllm: An efficient and effective multi-modal extension of large language models for text-rich document understanding*, 2024. arXiv: 2408.15045 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2408.15045

# Appendix A: Evaluation Schemas

A recurring challenge with existing document extraction benchmarks is incomplete or ambiguous schema documentation. Without clear specifications, it can be difficult to understand expected field formats, handling of optional values, or normalization rules. Researchers attempting to reproduce results must often reverse-engineer these details from examples or evaluation scripts, leading to inconsistent implementations and incomparable metrics.

To improve reproducibility, we provide explicit schemas for claim incidents and the main heterogeneous row families that benefit most from a public field contract. Other operations families are defined by their ground-truth field contracts and scored by the generic record-list evaluator. The evaluation scripts normalize model outputs before scoring so format errors are caught early rather than silently degrading metrics.

## A.1 Financial Breakdown Schema

Each incident contains four financial breakdown objects (`bi`, `pd`, `lae`, `ded`) with the following fields:

| Field | Type | Default | Description |
|---|---|---|---|
| `reserve` | float | 0.0 | Amount reserved for potential payout |
| `paid` | float | 0.0 | Amount already paid |
| `recovered` | float | 0.0 | Amount recovered (e.g., deductible) |
| `total_incurred` | float | 0.0 | Reserve + Paid - Recovered |

## A.2 Loss Run Incident Schema

The primary entity schema representing a single insurance claim incident:

| Field | Type | Default | Description |
|---|---|---|---|
| `incident_number` | string | required | Incident number (e.g., #12345) |
| `reference_number` | string | required | Reference ID (e.g., L240123) |
| `company_name` | string | required | Trucking company name |
| division | string | General | Company division |
| `coverage_type` | string | required | Coverage type (Liability, Physical Damage, Inland Marine, Cargo) |
| status | string | required | Open or Closed |
| `policy_number` | string | required | Policy identifier |
| `policy_state` | string | required | Policy state abbreviation |
| `cause_code` | optional string | null | Internal cause code |
| description | string | required | Detailed incident description |

| Field | Type | Default | Description |
|---|---|---|---|
| handler | string | Claims Adjuster | Claims handler |
| `unit_number` | optional string | null | Vehicle/truck unit ID |
| `date_of_loss` | string | required | Date incident occurred |
| `loss_state` | string | required | State where loss occurred |
| `date_reported` | string | required | Date reported to insurance |
| agency | optional string | null | Insurance agency name |
| insured | string | required | Insured party name |
| claimants | list of strings | [] | List of claimants |
| `driver_name` | optional string | null | Driver name at time of incident |
| bi | financial breakdown | {} | Bodily Injury |
| pd | financial breakdown | {} | Property Damage |
| lae | financial breakdown | {} | Loss Adjustment Expense |
| ded | financial breakdown | {} | Deductible |
| `adjuster_notes` | optional string | null | Additional adjuster notes |

## A.3 Extraction Output Shape

Claim/loss-run tasks return incident records:

| Field | Type | Default | Description |
|---|---|---|---|
| incidents | list of LossRunIncident | required | List of extracted incident records |

Other long-list tasks return generic records:

| Field | Type | Default | Description |
|---|---|---|---|
| records | list of objects | required | List of extracted target records using the fields specified for that document family |

The repository publishes standalone JSON Schema files for `loss_run_incident`, `loss_run_claim_row`, `ifta_multisection_jurisdiction_row`, `policy_schedule_record`, and `policy_packet_item`. The multisection IFTA schema requires each jurisdiction row to

include return context (`return_id`, year, quarter, filing type), Schedule A mileage/gallon fields, and Jurisdictions tax-detail fields.

## A.4 Extraction Prompt Template

Prompt-based claim multi-hop incident extraction uses the core template shown below. The evaluation code substitutes `{schema_json}` with the JSON schema for the extraction object and `{ocr_text}` with the selected transcript.

```
Extract all incident records from the following document.

Requirements:
- Return ALL incidents you can find in the document.
- Each incident MUST include ALL fields in the schema.
- When a value is unknown:
  - For required string fields, use "" (empty string), not null.
  - For optional fields, use null.
  - For list fields, use [].
  - For numeric fields, use 0.0.
- Output MUST be valid JSON that conforms to the schema.

Schema (JSON Schema):
{schema_json}

Output JSON shape:
{
  "incidents": [ ... ]
}

Document:
{ocr_text}
```

For generic operations and policy rows, the runner derives a sample-specific field contract from the keys and record groups present in the ground-truth object structure. This is schema disclosure: the contract contains field and group names but no target values, target counts, or ground-truth files. The output shape is:

```
{
  "records": [ ... ]
}
```

## A.5 Agentic Sandbox Instructions

The released Codex and Claude Code runners use the same task prompt. The temporary workspace contains only the OCR transcript, a field contract, the prompt, and an empty output directory. A macOS profile denies access to the benchmark repository, and the prompt prohibits using other host files. For claim tasks, the output path is `output/incidents.json`; generic tasks use `output/records.json`.

```
You are dispatched as a subagent for a specific extraction task; skip
superpowers skills.

Your workspace contains:
- document_ocr.md: OCR transcript of one benchmark PDF.
- field_contract.md: the target output shape and allowed fields.

The official ground truth is not present. Do not use files outside this
workspace.

Task:
Extract the complete target list from document_ocr.md according to
field_contract.md.
Write valid JSON to output/incidents.json.

You may inspect the document, write temporary scripts, and verify counts.
Before finishing:
- Confirm the JSON parses.
- Spot-check beginning, middle, and end records against the OCR.
- Ensure field names match field_contract.md exactly.
- Ensure every target row or target record type requested by the contract is
  represented.

Do not include explanations in the JSON file.
```

The field contract contains the claim JSON Schema or the generic record groups and allowed fields described in Appendix A.4. It contains no record values or target counts.

## A.6 Field Scoring Rules

We compute strict normalized-record completeness first, then schema-conformant field precision/recall/F1 as partial-credit diagnostics.

**Step 1: Canonicalize each incident (schema validation + normalization).** Both predictions and ground truth are parsed as full `LossRunIncident` objects and then normalized:

- **Incident identifier**: common prefixes in `incident_number` (e.g., `Incident #`, `#`, `INC`) are stripped; whitespace and case are normalized.
- **String fields**: all strings are trimmed and compared case-insensitively. For optional string fields, the empty string is treated as `null`. Optional string fields are:
  - `cause_code`
  - `unit_number`
  - `agency`

  - `driver_name`
  - `adjuster_notes`

- **Dates**: supported date forms are normalized to `MM/DD/YYYY`.
- **Claimants**: coerced to a list; each entry is trimmed and case-normalized; empty entries are dropped; the list is sorted.
- **Financial breakdowns**: for each breakdown object (`bi`, `pd`, `lae`, `ded`), we ensure it is an object and normalize each numeric field by converting to float, rounding to two decimals, and mapping negative zero to zero. Unparseable values are treated as 0.0. Zero-valued default breakdown fields are not scored. Breakdown numeric fields are:
  - `reserve`
  - `paid`
  - `recovered`
  - `total_incurred`

Generic records are canonicalized recursively. Blank fields are omitted; strings are whitespace-normalized and compared case-insensitively; dates, decimals, percentages, currency values, and accounting-parenthesis negatives are canonicalized; and list values are order-normalized. State and jurisdiction names normalize to two-letter codes, fuel descriptions to parenthetical codes, policy line-of-business names to BOP/WC/CGL acronyms, and `Applies within` clause-scope wrappers to the core scope phrase. A visible `Unit` prefix is removed from vehicle identifiers, and `Quarter Return` is treated as a heading alias for `Quarterly Return`; extra heading context is retained and therefore remains an error. For field diagnostics, exact records are anchored first and remaining records are matched deterministically by overlapping non-global field pairs. The `record_type` field, hidden row identifiers, and repeated document-level policy fields are excluded from field matching and field-pair scoring. Strict exact-record comparison uses the complete normalized public record.

**Step 2: Compare complete normalized records.** Let $R_G$ and $R_P$ be multisets of normalized ground-truth and predicted records. Exact-record recall is $|R_G \cap R_P|/|R_G|$. A document is complete only when $R_G = R_P$, so missing records, extra records, wrong fields, and duplicate multiplicity all affect strict completeness. Record-list order is not scored in this release.

**Step 3: Flatten incidents into a multiset of field-value pairs.** For each canonicalized incident, we emit a list of triples (`incident_id`, `field_path`, `value`). Nested financial fields are represented with dotted paths (e.g., `bi.reserve`).

**Step 4: Compute partial-credit field diagnostics.** Let $\mathcal{F}(G)$ be the multiset of flattened triples from the ground truth and $\mathcal{F}(P)$ the multiset from predictions. We compute

$$\text{found} = |\mathcal{F}(G) \cap \mathcal{F}(P)|, \tag{7}$$

$$\text{recall} = \frac{\text{found}}{|\mathcal{F}(G)|}, \tag{8}$$

$$\text{precision} = \frac{\text{found}}{|\mathcal{F}(P)|}, \tag{9}$$

$$\text{F1} = \frac{2\,\text{precision}\,\text{recall}}{\text{precision} + \text{recall}}. \tag{10}$$

The multiset formulation means that if a document contains exact duplicate incidents (same normalized `incident_id`), they are counted with multiplicity, and the score only credits matches up to the minimum count in each multiset.

**Additional diagnostics.** We also report:

- **Missing/extra incident IDs**: computed from the set of normalized incident IDs present in each list.
- **Exact-record precision and $F_1$**: computed from the same normalized-record intersection when systems over-extract.
- **Evaluation-role and stressor slices**: aggregated from the public family and stressor metadata.

## A.7 Corpus Inventory and Stressor Map

Table 10 lists the technical document families used by the manifest. It is an inventory, not a difficulty ordering.

Table 10: Current LongListBench corpus by technical family.

| Document family | PDFs | Target records |
|---|---|---|
| `ifta_mileage_by_vehicle` | 8 | 17,565 |
| `ifta_multisection_return_packet` | 2 | 796 |
| `ifta_return_schedule_details` | 3 | 2,737 |
| `ifta_tax_return_summary` | 2 | 1,520 |
| `driver_mvr_request_and_roster` | 3 | 1,260 |
| `loss_run_external` | 3 | 900 |
| `vehicle_schedule_spreadsheet_export` | 2 | 1,600 |
| `ifta_tax_return_inquiry_detail` | 2 | 1,300 |
| `driver_schedule_spreadsheet_export` | 1 | 500 |
| `claim_crosspage_multihop` | 3 | 77 |
| `policy_multihop_bop` | 1 | 344 |
| `policy_multihop_wc` | 1 | 438 |

| Document family | PDFs | Target records |
|---|---|---|
| policy_multihop_cgl | 1 | 562 |
| **Total** | **32** | **29,599** |

Table 11: Representative pages for auditing canonical stressors. Page numbers refer to released PDFs.

| Stressor | Example PDF pages | Visible document evidence |
|---|---|---|
| page_breaks | ifta_mileage_by_vehicle_001, pages 2–3 | A repeated list continues with inherited unit context; this is pagination pressure, not an item-level split. |
| split_records | ifta_multisection_return_001, pages 2 and 4 | One jurisdiction record combines mileage and gallon fields with later tax-detail fields. |
| multi_row | loss_run_external_001, pages 1–2; driver_mvr_packet_001, page 10 | Records include description rows and driver or MVR detail blocks. |
| duplicates | loss_run_external_001, pages 1–2; multihop_bop_012_001, pages 45–68 | Summary, archived, or prior-term material creates near-duplicate distractors. |
| large_doc | ifta_mileage_by_vehicle_008; mixed_cgl_040_001 | The files reach 76 and 133 pages. |
| multiple_tables | ifta_tax_inquiry_001, page 1; loss_run_external_001, pages 1–2 | Targets appear beside support tables, empty tables, summaries, and totals. |
| multi_column | mixed_cgl_040_001, pages 67–86; multihop_wc_025_001, pages 54–73 | Policy provisions use two-column form pages. |
| merged_cells | loss_run_external_001, page 1; ifta_mileage_by_vehicle_002, page 2 | Section-spanning rows interrupt regular table structure. |
| ocr_condition | Any PDF and matching file under data/transcripts/ocr_gemini/ | Released text is OCR from page images, not HTML source text. |
| ocr_layout_condition | ifta_multisection_return_001, pages 2 and 4 | OCR preserves two visually distinct tables rather than a clean joined row. |
| long_range_evidence | multihop_012_001_crosspage, pages 3, 26, 28, 46, 47, 60; mixed_040_001_crosspage, pages 3, 50, 54, 96, 97, 137 | A front claim row joins to distant driver, policy, cause, claimant, and ledger sections. |

| Stressor | Example PDF pages | Visible document evidence |
|---|---|---|
| cross_section_join | ifta_multisection_return_001, pages 1, 2, 4 | A target combines return context, mileage/gallons, and tax-detail values. |
| repeated_keys | ifta_multisection_return_001, pages 2, 4, 8, 10 | Jurisdiction codes recur across returns, so state alone is not a unique key. |
| heterogeneous_record_list | multihop_bop_012_001, pages 2–17 and 39–90; mixed_cgl_040_001, pages 2–25 and 54–133 | One list mixes locations, coverages, forms, endorsements, premiums, and clauses. |