FINSTRUCT: LLM-Based Logical Structure Recovery for Financial Filings

Mazen Ahmed[1,2] Yunfei Long[3] Michael Kampouridis[1]
Rachel Tattersall[2] Kimberley Clague[2]

[1]School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
[2]Cooper Parry, UK
[3]Queen Mary University of London, London, UK

mm24829@essex.ac.uk yunfei.long@qmul.ac.uk mkampo@essex.ac.uk
rachelt@cooperparry.com kimberleyc@cooperparry.com

Abstract

Financial-filings workflows depend on more than extracting reported values: each number must be placed in the document context that makes it reviewable, comparable, and auditable. In practice, this logical structure including the relevant section, note, heading, row, column, period, currency, and surrounding text—is often clear to professional reviewers but only partially encoded in inline XBRL markup. As a result, rule-based parsers can work on familiar layouts yet fail silently when document structure departs from expected templates. We present **FINSTRUCT**, an on-premise LLM system designed for confidential financial-document workflows. **FINSTRUCT** runs locally on a single 16 GB GPU, requires no task-specific fine-tuning, predicts nine structural fields for each reported fact, and provides reviewer-facing rationales together with repeated-sample agreement signals for human-in-the-loop review. On 1,493 manually annotated facts from five UK FRS 102 filings produced by a single accounts-production stack, **FINSTRUCT** raises macro accuracy from 50.6% to 74.8% and micro accuracy from 63.1% to 85.1% over a deterministic parser. More importantly for deployment, it recovers structure at 83.5% in accuracy on a filing where the rule-based parser silently misses whole sections, and its agreement signal separates more reliable predictions from cases requiring review. These results position logical structure recovery as a practical document-understanding layer for regulated financial automation, rather than a conventional parsing problem alone.

1 Introduction

Financial filings are increasingly processed by automated systems, but reliable automation requires more than extracting numbers. In financial services, a value is useful only when placed within the document structure that gives it meaning: the relevant statement or note, row and column, reporting year, currency context, and surrounding explanation. The same value may denote a current-year balance, prior-year comparative amount, or subtotal depending on its position in the rendered filing. Human reviewers infer this context from headings, table layout, typography, and nearby prose, whereas automated systems must recover it explicitly. This is difficult because much of the relevant structure is visible to readers but only partially encoded in filing markup. Inline XBRL records facts, values, accounting concepts, periods, and units, but often omits the logical position needed to interpret a value. As a result, rule-based and layout-driven parsers can work on expected templates yet become brittle when structure must be inferred, sometimes failing silently across whole regions of a filing. In regulated financial workflows, such layout-dependent collapse is a deployment risk.

Prior work explains this gap. Document-understanding methods recover visual and geometric structures such as tables, cells, rows, columns, and layout-aware representations (Adiga et al., 2019; Kasem et al., 2024; Xu et al., 2020, 2021)—but they do not necessarily identify the logical context that governs a reported value. Meanwhile, Financial NLP has advanced XBRL tagging, numerical reasoning, intent-relevant extraction, and retrieval over financial reports (Loukas et al., 2022; Chen et al., 2021; Sarker and Rai, 2025; Li et al., 2025), often assuming that such structure is already available. While LLM-based extraction and in-context learning offer adaptation without parameter updates (Brown et al., 2020; Xie et al., 2023; Xu et al., 2024), most existing work operates over short or pre-segmented spans and fails to address confidential, full-filing deployment. We target this intermediate layer: recovering the logical structure needed before downstream financial automation can reliably use reported values.

We frame logical structure recovery as a document-understanding task under production

constraints. The system must operate on confidential client filings, run on modest local hardware, avoid task-specific fine-tuning because no labelled training data exists, and produce outputs that are inspectable by professionals. To address these requirements, we present FINSTRUCT, an LLM-based system for recovering the logical structure of financial filings. For each anchor-eligible reported fact, FINSTRUCT predicts nine structural fields: section title, note label, sub-heading, row label, column header, year, currency label, and two surrounding-context fields. The system supports three operating modes: a deterministic rule-based extractor, a zero-shot LLM extractor, and an in-context LLM extractor. The LLM modes share a two-stage design that first resolves the document section and then recovers the remaining fields conditioned on that section. FINSTRUCT runs locally on a single 16 GB GPU, requires no task-specific fine-tuning, and provides reviewer-facing rationales together with repeated-sample agreement signals for human-in-the-loop review. Building on self-consistency (Wang et al., 2023), we use repeated sampling not only to aggregate predictions but also to rank predictions by reliability for review routing.

We evaluate FINSTRUCT on a self-built benchmark of 1,493 manually annotated facts from five UK FRS 102 filings produced by a single accounts-production stack. To our knowledge, this is the first dataset of its kind for logical structure recovery in financial filings. C3 raises macro accuracy from 50.6% to 74.8% and micro accuracy from 63.1% to 85.1%. Beyond aggregate gains, FINSTRUCT recovers structure with 83.5% accuracy on a filing where the rule-based parser silently fails across whole sections, demonstrating resilience to layout-dependent breakdowns. Its agreement signal further supports risk-aware deployment by separating reliable predictions from cases requiring human intervention while preserving an auditable review trail.

2 Task Formulation and System Design

2.1 Task Formulation

A UK statutory filing in inline XBRL has two simultaneous representations: a rendered report read by humans and a machine-readable layer of embedded facts. We formulate the recovery task at the fact level. Given an anchor-eligible fact and its rendered neighbourhood in the filing, the system must recover the logical position that a professional reader would use to interpret the reported value. The embedded XBRL fact records its value, accounting concept, period context, and unit, but it does not, in general, encode where that occurrence sits in the report's logical structure. For example, a concept tag may identify the number 98,139 as intangible assets without indicating that this particular occurrence is the current-year net book value of computer software in the intangible-assets note. Such placement is visible on the rendered page but may be absent from the markup: section boundaries can appear only as typographic changes, note headings and column headers can share generic markup containers, and multi-page notes may be connected only through visual continuity (Figure 4).

We define logical structure recovery as the task of assigning each anchor-eligible fact a structured record that captures its reading position in the filing. The output schema contains nine fields: `section title`, selected from a closed set of fourteen statement or report sections; `note label`; `sub heading`; `row label`; `col header`; `year`; `currency label`; and two surrounding-context fields, `text before` and `text after`. These fields are operational rather than merely descriptive: the same value and accounting concept may recur across different rows, years, notes, and statements, and only the recovered structure distinguishes these occurrences for review, comparison, and automated checking. The task therefore requires not only accurate field recovery, but also a signal of uncertainty so that low-confidence predictions can be routed to human review.

2.2 Production constraints

Three production constraints shape the design of FINSTRUCT. **Confidentiality** requires all inference to run inside the firm's controlled environment: client filings are confidential, which motivates an on-premise system that can operate on modest local hardware. **Auditability** requires that each prediction be inspectable by a professional reviewer and traceable to the document context from which it was produced. This motivates a fixed output schema, persisted model outputs, reviewer-facing rationales, and confidence signals that support review rather than unconditional automation. **Data scarcity** rules out supervised training as the solution: no labelled corpus exists for this level of logical structure, and creating one requires expert review under the same confidentiality restrictions.

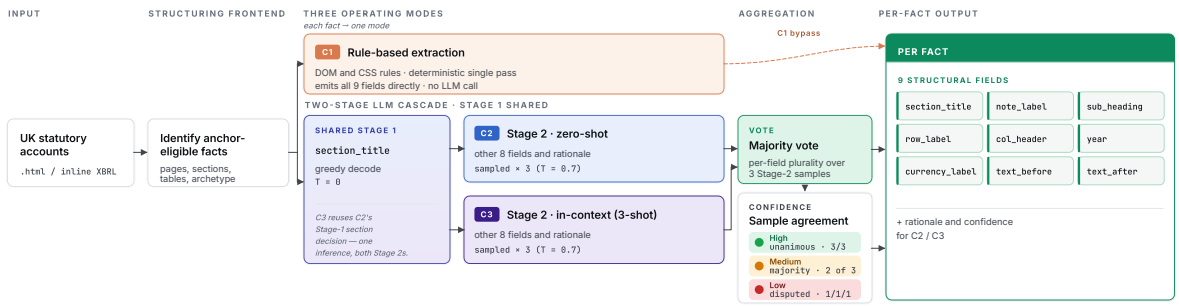


Figure 1: The FINSTRUCT pipeline. A structuring frontend assigns each anchor-eligible fact an archetype; **C1** emits all nine fields and bypasses aggregation, while **C2/C3** share a two-stage cascade (Stage 1 predicts `section_title`, Stage 2 the other eight). C3 reuses C2's Stage 1 with three leave-one-filing-out demonstrations; Stage 2 is sampled three times, and agreement over the samples gives a confidence signal for human-review routing.

We therefore design FINSTRUCT to use deterministic rules, zero-shot prompting, and in-context demonstrations without task-specific fine-tuning.

2.3 The FINSTRUCT system

FINSTRUCT compares three recovery modes over the same fact-centred input and nine-field output schema. This shared setup ensures that differences between modes reflect the recovery strategy itself, rather than changes in task definition, context construction, or output format. Before extraction, a structuring frontend segments the rendered filing into pages, sections, tables, headings, and local text regions. Each visible anchor-eligible fact is linked to its neighbourhood, assigned an archetype (`table numeric`, `table text`, `narrative block`, or `standalone`), and represented with the contextual cues a reader would use: rows, columns, headings, period and currency labels, and surrounding prose. Hidden facts are excluded because they lack a visible reading position. The three modes instantiate progressively stronger assumptions about how logical structure can be recovered.

- C1 represents the deterministic boundary of conventional parsing: it uses DOM structure, CSS cues, table geometry, heading proximity, and archetype-specific rules to assign all nine fields in a single pass. This mode is appropriate when logical structure is explicitly encoded or follows anticipated layout conventions, but it is expected to degrade when the relevant evidence is distributed across visual layout, headings, and prose.

- C2 treats the same task as constrained language understanding. Instead of predicting the full schema jointly, it uses a two-stage cascade: Stage 1 resolves `section title` from a closed inventory of fourteen sections, and Stage 2 predicts the remaining eight fields with a section-specific prompt and schema-valid JSON output. This decomposition reflects the hierarchy of financial filings, where the section acts as a high-level interpretive anchor that constrains note labels, headings, rows, columns, and contextual text. It also makes the recovery process more inspectable by separating document-level localisation from field-level interpretation.

- C3 keeps the C2 architecture but conditions Stage 2 on retrieved in-context demonstrations. Examples are selected under a leave-one-filing-out regime, prioritising the same section and archetype as the target fact. This tests whether limited expert-labelled evidence can improve recovery while preserving the production constraints of local inference, no task-specific fine-tuning, and no transfer of client data to external services. Since C2 and C3 share the same input representation, section decision, output schema, and inference protocol, their comparison isolates the contribution of in-context demonstrations.

For the LLM modes, FINSTRUCT also derives an agreement-based review signal from repeated Stage 2 sampling. Each field is sampled three times and assigned an agreement label: `unanimous` if all samples match, `majority` if two match, and `disputed` if all differ. The final prediction is selected by plurality voting, while the agreement label is retained as an indicator of reliability. We do not interpret this signal as a calibrated probability; rather, it operationalises a central requirement

	F1	F2	F3	F4	F5
Anchor-eligible facts	534	382	236	136	205

Table 1: Anchor-eligible facts per filing, giving 1,493 fact-level evaluation instances in total.

of regulated document workflows, where uncertain predictions should be surfaced for professional review instead of being silently accepted. Thus, the system is designed not only to recover logical structure, but also to support auditable human-in-the-loop deployment.

3 Data and Evaluation Methodology

3.1 Corpus

We evaluate FINSTRUCT on five real UK statutory accounts filings prepared under FRS 102, including Section 1A accounts, produced by a single mainstream accounts-production tool and submitted as inline XBRL. The filings are genuine client documents processed under the confidentiality constraints in Section 2.2: entity names, addresses, and reported values were anonymised, while rendered structure was preserved for valid logical-structure annotation. This deployment-derived corpus is a focused benchmark rather than a broad sample of UK filings, yielding 1,493 anchor-eligible evaluation instances across the main structural forms observed in production filings. From 1,793 inline XBRL facts, we exclude continuation tails and hidden facts because they lack independent, visible reading positions. All nine structural fields were manually annotated by two domain experts reading the rendered filings directly, producing gold labels independent of parser or model outputs. Both annotators labelled all 1,493 facts independently; per-field inter-annotator agreement is reported in Appendix D, with overall agreement of 97.7% and macro Cohen's κ of 0.933. Residual disagreements were resolved by the senior annotator to produce the final gold set. The five filings come from five distinct entities operating in different industry sectors rather than repeated filings by one entity, so C3's leave-one-filing-out demonstrations are drawn from other companies and its gain reflects cross-entity transfer rather than adaptation to a single preparer's house style. Two annotation conventions affect scoring and are described in Appendix C.

3.2 Matching and metrics

We score predicted fields with field-specific criteria: normalised exact match for most fields, Jaro–Winkler similarity at a fixed 0.85 threshold for free text, and sentinel-aware matching for context fields where absence is a valid label. Correct matches include exact, fuzzy, and sentinel matches; missing predictions are incorrect unless absence is gold-labelled by sentinel. We report per-field accuracy, macro accuracy over nine fields, and pooled micro accuracy, with 95% percentile bootstrap confidence intervals over facts. Because modes share the same facts, pairwise field-level comparisons use McNemar's exact test with Holm–Bonferroni correction over 27 comparisons. Full details appear in Appendix C.

3.3 Models and inference

The primary LLM configuration uses Qwen3-14B, AWQ-quantised and served locally with vLLM on a single 16 GB GPU. Stage 1 is decoded greedily with $T{=}0.0$, while Stage 2 is sampled with $T{=}0.7$ and $N{=}3$ samples per fact. No model is fine-tuned, and no client data leaves the local environment. To test whether the observed pattern is specific to the primary model, we also evaluate a second open-weight instruction-tuned model, Llama-3.1-8B-Instruct, as a robustness check; these results are reported in the Appendix.

4 Experimental Results and Analysis

4.1 Overall performance

Table 2 reports per-field accuracy for the three operating modes. The results show a clear progression from deterministic parsing to LLM-based recovery. Averaged across the nine fields, C1 reaches 50.6% macro accuracy, while C2 improves this to 71.0% and C3 further raises it to 74.8%. The same pattern appears under micro accuracy, which increases from 63.1% for C1 to 80.4% for C2 and 85.1% for C3. In relative terms, C3 reduces macro-level error by nearly half compared with the rule-based baseline. These gains indicate that a substantial portion of the structure required to interpret reported facts is not reliably recoverable through deterministic parsing alone. The improvement is achieved without task-specific fine-tuning and under the on-premise constraints described in Section 3.3. This is important for deployment: the result does not depend on sending confidential filings to a hosted model or on constructing a large labelled training

Field	C1 [lo, hi]	C2 [lo, hi]	C3 [lo, hi]
section_title	86.5 [84.7, 88.1]	99.7 [99.3, 99.9]	99.7 [99.4, 99.9]
note_label	60.0 [56.9, 63.9]	86.6 [84.2, 88.8]	93.3 [91.4, 94.9]
sub_heading	3.4 [2.2, 4.9]	55.2 [51.7, 58.8]	72.7 [69.3, 75.7]
row_label	65.2 [62.3, 68.0]	82.1 [79.9, 84.2]	85.2 [83.2, 87.2]
col_header	60.9 [58.1, 63.7]	80.6 [78.3, 82.9]	82.1 [80.0, 84.4]
year	85.4 [82.9, 87.8]	73.9 [70.9, 76.6]	84.0 [81.5, 86.2]
currency_label	82.8 [80.6, 84.8]	95.8 [94.6, 96.9]	96.1 [94.9, 97.1]
text_before	6.1 [3.8, 9.2]	34.7 [29.0, 40.3]	28.5 [22.7, 34.3]
text_after	5.3 [2.8, 8.1]	30.8 [25.6, 36.0]	31.8 [25.9, 37.3]
Macro	50.6 [49.4, 51.9]	71.0 [69.6, 72.5]	74.8 [73.4, 76.3]
Micro	63.1 [61.3, 64.8]	80.4 [79.1, 81.6]	85.1 [83.9, 86.2]

Table 2: Per-field accuracy (%) on Qwen3-14B-AWQ with 95% percentile bootstrap CIs (1,000 resamples, fact-level). Denominators exclude pairs where both gold and prediction are absent. Best per row is not bolded here because differences are read against the CIs.

Section	n	C1	C2	C3
NOTES	669	57.8	73.3	76.8
DPL_SCHEDULES	222	75.6	91.7	96.2
BALANCE_SHEET	169	62.4	88.7	91.6
SOCIE	119	72.9	96.5	97.1
INCOME_STATEMENT	87	72.5	83.9	96.5
COMPANY_INFO	62	38.9	100.0	97.6
DIRECTORS_REPORT	59	32.6	60.1	77.4
DPL	56	93.6	82.6	95.2
AUDITORS_REPORT	33	21.4	69.5	70.8
COVER	15	82.5	65.0	88.1
STRATEGIC_REPORT	2	25.0	66.7	100.0

Table 3: Per-section accuracy (%), Qwen3-14B-AWQ, micro-averaged over the nine fields (n = facts per section). DPL is the only section where C2 trails C1; C3 recovers it. STRATEGIC_REPORT (n=2) is a degenerate denominator.

set. Instead, the LLM modes recover additional structure from the same fact-centred context available to the rule-based system.

Because facts cluster within filings, the fact-level intervals in Table 2 understate uncertainty about how these results transfer to a new filing. We therefore recompute all intervals by resampling filings rather than facts (n=5, 1,000 resamples). The intervals widen substantially: macro accuracy C1 50.6 [28.7, 60.6], C2 71.0 [63.0, 75.7], C3 74.8 [69.0, 78.8]. The C1–C3 intervals remain disjoint under clustering, while C2 and C3 overlap, so we do not rest the C2–C3 distinction on fact-level intervals alone. A per-filing breakdown, which makes no distributional assumption, shows C3 exceeding both C1 and C2 on every one of the five filings independently, under both macro and micro accuracy; the aggregate gain is therefore not driven by a single favourable document.

4.2 Field-level and section-level behaviour

The aggregate improvement masks important differences across fields. Fields that are often explicit in the document representation, such as `section title`, `currency label`, and `row label`, already perform relatively well under C1, leaving less room for improvement. In contrast, context-dependent fields show much larger gains. The most notable example is `sub heading`, which increases from 3.4% accuracy under C1 to 72.7% accuracy under C3. The surrounding-context fields, `text before` and `text after`, also improve substantially.

The `year` field highlights the complementarity of the approaches. C1 performs strongly because period information is often explicitly encoded, while C2 underperforms and C3 recovers much of the gap through in-context examples. This suggests that deterministic parsing remains effective for regular, well-structured information, whereas LLM-based extraction is most valuable when structure must be inferred from context.

The section-level results in Table 3 show the same pattern. C3 improves over C1 across most sections, with the largest gains in report and narrative sections where logical structure is conveyed through headings and prose rather than explicit markup. Overall, the results reinforce that logical structure recovery is a broader document-understanding task rather than solely a table-parsing problem.

4.3 Silent rule failure and graceful recovery

The average accuracy hides the most critical failure in the deployment. In filing F5, C1 produces no output for 162 facts while maintaining full coverage on the other four filings. The missing facts span whole statement and report sections, indicat-

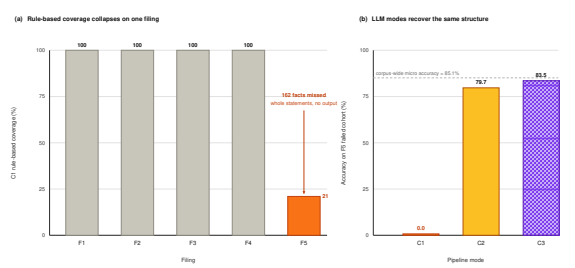


Figure 2: Catastrophic rule failure on F5. (a) C1 produces no output for 162 facts across whole sections, while coverage is complete on F1–F4. (b) On that cohort C1 scores 0%; C2 and C3 recover to 79.7% and 83.5% (dashed = corpus micro accuracy).

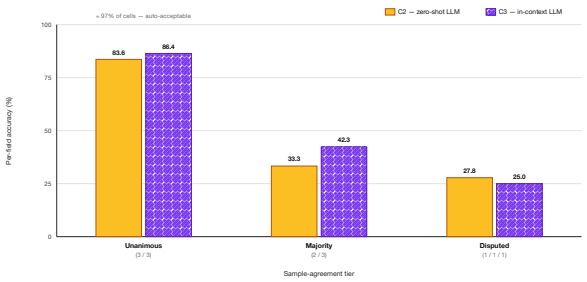


Figure 3: Per-field accuracy by sample-agreement tier (C2, C3). Accuracy falls from unanimous to disputed, so agreement ranks predictions by reliability; we use it as a triage signal for targeted review, not a guarantee.

ing a systematic layout-dependent collapse rather than isolated errors. The cause is a DOM structure variation present in F5 but not in F1–F4: its page-level container markup differs in shape, and C1's heading extraction does not recover headings under that variant. Because section identity anchors the remaining eight fields, the failure propagates: section detection collapses and the dependent fields return null across the affected regions. C1 raises no error at any point, which is why the failure is silent and surfaced only under coverage analysis. In addition, the LLM modes do not exhibit the same failure pattern. In the 162-fact cohort where C1 does not produce output, C2 and C3 recover structure with pooled accuracies of 79.7% and 83.5%, respectively. This result is important because it reflects a practical production risk. The variant is addressable by extending the rule set, but that is the point rather than a caveat to it: filings are produced by many tools, versions and preparer conventions, each introducing markup variation the rules must anticipate in advance, and the failure gives no diagnostic signal when they do not. The evidence comes from one out-of-distribution filing, so it should not be read as estimating the frequency of such failures. It nevertheless demonstrates a failure mode that aggregate accuracy can hide and motivates reporting per-filing coverage alongside accuracy in deployment evaluations.

4.4 Agreement as a confidence signal

For C2 and C3, repeated Stage 2 sampling provides a useful reliability signal. As shown in Figure 3, agreement is strongly associated with correctness. Under C3, fields for which all three samples agree are 86.4% in accuracy, while fields with only a two-of-three majority are 42.3% in accuracy and fields with three different predictions are 25.0% in accuracy. C2 shows the same monotonic pattern. Agreement therefore ranks predictions by reliability, even though it is not calibrated as a probability.

This distinction matters for regulated workflows. The agreement signal should not be used to justify unconditional automation: even unanimous predictions are not always correct. Its value is instead operational. A cell-level policy that auto-accepts only unanimous fields accepts 96.95% of cells at 86.4% accuracy, with the routed remainder at 42.0%. A stricter fact-level policy that routes any fact containing a non-unanimous field accepts 87.81% of facts at 87.5% accuracy, sending 182 of 1,493 facts (12.2%) to review. C3 also reaches unanimity more often than C2 (96.9% versus 93.8% of scored cells), so the more accurate mode is also the more decisive one. These figures characterise the evaluation set rather than validated policies, as no held-out split was used: the intended use is that a firm selects its own operating point against its risk tolerance rather than adopting a threshold we recommend. In both cases the system converts uniform manual checking into targeted review, allowing professional attention to focus on less reliable predictions while retaining an auditable decision trail.

4.5 Robustness and deployability

To assess robustness, we repeat the LLM experiments with Llama-3.1-8B-Instruct and observe the same qualitative pattern: C3 consistently outperforms C2, indicating that the benefit of in-context demonstrations is not tied to a single model. We therefore treat Qwen3-14B as the primary configuration and report additional model results in Appendix I.

The smaller configurations also serve as a stress test. Zero-shot extraction falls below the rule-based baseline both for Llama-3.1-8B-Instruct (macro 50.1 versus C1's 50.6) and for Qwen3-1.7B (42.1),

and in-context demonstrations recover above it in both cases. Zero-shot LLM extraction is therefore not by itself sufficient at modest capacity; demonstrations are what clear the deterministic baseline.

The primary deployment configuration runs locally on a single 16 GB GPU, with both C2 and C3 requiring approximately 11 seconds per fact. Local inference is driven primarily by confidentiality requirements. At the same time, the on-premise setup avoids recurring hosted-inference costs and keeps all prompts, outputs, rationales, and logs within the firm's controlled environment.

Overall, the results support three conclusions. First, LLM-based recovery substantially improves the reconstruction of the logical structure in financial statements, particularly where the structure is implicit rather than explicitly encoded. Second, repeated-sample agreement provides a practical mechanism for routing uncertain cases to human review. Third, beyond improved headline performance, FINSTRUCT offers resilience to silent layout-dependent parser failures while compatible with confidentiality and auditability requirements.

5 Conclusion

This paper introduced FINSTRUCT, an on-premise system for recovering the logical structure of financial filings, addressing the problem that reported values in inline XBRL cannot be reliably used for review, comparison, or automated checking unless they are located within the document context that gives them meaning: section, note, heading, row, column, period, currency, and surrounding text. We argued that this context is often apparent to human readers but only partially represented in machine-readable markup, making logical structure recovery a document-understanding problem rather than a purely rule-based parsing task. Across 1,493 manually annotated facts from five UK FRS 102 filings produced by a single accounts-production stack, FINSTRUCT substantially improves over deterministic extraction, raising macro accuracy from 50.6% to 74.8% and micro accuracy from 63.1% to 85.1% without task-specific fine-tuning. The results also show why this improvement matters in deployment: the rule-based mode fails silently across whole sections of one unfamiliar filing, whereas the LLM modes recover structure for the same cohort with high accuracy. In addition, repeated-sample agreement provides a practical reliability signal for routing uncertain predictions to professional review rather than silently accepting them. The contribution of FINSTRUCT therefore is not only greater extraction accuracy, but also a deployment-compatible formulation of the recovery of logical structures under confidentiality, auditability, and data-scarcity constraints. Although our evaluation focuses on UK inline XBRL filings, the broader problem extends to many business and regulatory documents whose logical structure is designed for human comprehension but only weakly encoded for machines.

Limitations

Our evaluation is based on five FRS 102 filings from a single accounts-production stack. Although they yield 1,493 densely annotated facts across the main structural archetypes, the corpus is a focused deployment benchmark rather than a broad sample of UK filings, and we do not assess performance across other preparers, formats, standards, or document-generation tools. The silent-failure analysis also relies on one out-of-distribution filing, showing that wholesale rule failure can occur but not how frequently. Gold labels were produced by two domain experts working independently, with agreement reported in Appendix D; both annotators work within the same firm and share its house conventions for reading statutory accounts, so correlated annotation bias may remain undetected. The LLM rationales are used only as reviewer-facing audit aids; we do not evaluate their faithfulness. Likewise, the agreement signal is predictive of correctness but not calibrated as a probability. Finally, as we evaluate one primary model and one secondary model, absolute performance should be read as model-dependent evidence for the approach rather than for any specific model.

Ethics

This work uses real client financial filings provided by the participating firm. All filings were processed under the firm's confidentiality requirements: entity names, addresses, and reported values were anonymised before research use, document structure was preserved without altering its meaning, and no client data left the firm's controlled environment at any stage. All inference runs on-premise; the system makes no calls to external services, which is a deliberate design requirement rather than only an efficiency choice. Although anonymisation substantially reduces disclosure risk, we do not

treat it as a guarantee against re-identification by an adversary with external knowledge.

FINSTRUCT is intended to support, not replace, professional judgement. It produces structural labels that feed a human-in-the-loop review process, and its confidence signal is designed to direct - not remove - that review. We make no claim that the system should operate without human oversight on regulated filings.

Acknowledgments

This work was supported by Innovate UK through Knowledge Transfer Partnership, a collaboration between Cooper Parry, the University of Essex, and Queen Mary University of London. We thank Cooper Parry's Business Process Optimisation, and tax teams for their domain expertise and annotation support.

References

Darshan Adiga, Shabir Ahmad Bhat, Muzaffar Bashir Shah, and Viveka Vyeth. 2019. Table structure recognition based on cell relationship, a bottom-up approach. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1–8, Varna, Bulgaria. INCOMA Ltd.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mahmoud Salaheldin Kasem, Abdelrahman Abdallah, Alexander Berendeyev, Ebrahem Elkady, Mahmoud Abdalla, Mohamed Mahmoud, Mohamed Hamada, Daniyar Nurseitov, and Islam Taj-Eddin. 2024. Deep learning for table detection and structure recognition: A survey. *ACM Computing Surveys*, 56(12):305:1–305:41. ArXiv:2211.08469.

Ying Li, Mengyu Wang, Miguel de Carvalho, Sotirios Sabanis, and Tiejun Ma. 2025. FinGEAR: Financial mapping-guided enhanced answer retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7239–7255, Suzhou, China. Association for Computational Linguistics.

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. 2022. FiNER: Financial numeric entity recognition for XBRL tagging. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4419–4431, Dublin, Ireland. Association for Computational Linguistics.

Soumick Sarker and Abhijit Kumar Rai. 2025. FinExBERT: User intent based text extraction in financial context using graph-augmented BERT and trainable plugin. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1402–1411, Suzhou, China. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*. ArXiv:2203.11171.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot NER with ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956, Singapore. Association for Computational Linguistics.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200. Association for Computing Machinery. ArXiv:1912.13318.

NOTES TO THE FINANCIAL STATEMENTS
FOR THE YEAR ENDED 30 JUNE 2025

11. Intangible assets

	Computer software £	Goodwill £	Total £
Cost			
At 1 July 2024	590,264	178,277	768,541
Additions	28,172	-	28,172
At 30 June 2025			
	618,436	178,277	796,713
Amortisation			
At 1 July 2024	416,544	178,277	594,821
Charge for the year	103,753	-	103,753
At 30 June 2025			
	520,297	178,277	698,574
Net book value			
At 30 June 2025	98,139	-	98,139
At 30 June 2024	173,719	-	173,719

(a) What a reader sees (rendered iXBRL filing, Note 11)

(b) What the markup encodes

```
<ix:nonFraction
  name="...IntangibleAssets"
  contextRef="c286" unitRef="u0"
  format="ixt:numcommadot">
    98,139
</ix:nonFraction>
```

The tag carries the value, accounting concept, period context, and unit — but not *where* the value sits in the report. (c) Logical context a reader infers, and **FINSTRUCT** recovers

section_title	Notes	
note_label	11	•
sub_heading	Net book value	•
row_label	At 30 June 2025	•
col_header	Computer software	•
year	2025	
currency_label	£	
text_before / _after	*(prose)*	

• = structural field with no direct counterpart in the markup.

Figure 4: The same reported value (boxed, **98,139**) in a UK statutory filing. From the rendered layout (a), a reader instantly infers the value's full logical context (c): section, note, sub-heading, row, column, period, and currency. The inline XBRL markup (b) records only the value, its accounting concept, period context, and unit; the fields marked • have no counterpart in the markup and must be inferred. **FINSTRUCT** recovers (c) from (b).

A Worked example

Figure 4 traces a single reported value through the three representations the task connects: what a reader sees in the rendered filing, what the inline XBRL markup encodes, and the logical context that FINSTRUCT recovers.

B Production constraints in detail

The three constraints summarised in Section 2.2 shape the system at a level of detail we expand here.

Confidentiality. Client filings are commercially sensitive and confidential. This encourages inference to run on-premise on hardware available to the firm, which places a practical upper bound on model size and motivates methods that avoid task-specific fine-tuning (which would itself require labelled client data to be assembled and stored). All inference in this work runs locally; no client data leaves the firm's environment at any stage.

Auditability. Because outputs feed a regulated workflow, every prediction must be inspectable by a reviewer, traceable to the document context that produced it, and reproducible as part of an evaluation or audit trail. This motivates per-prediction rationales, persisted raw model outputs, controlled sampling protocols, and explicit logging of model configuration, prompts, latency, and token usage. For each LLM call, FINSTRUCT persists the raw response, the parsed prediction, prompt metadata, latency, and token counts, so that failed, uncertain, or disputed predictions remain inspectable after the run.

Data scarcity. No gold standard exists for the logical structure of UK statutory filings at the level this task requires. Evaluation data therefore has to be created from scratch, under the same confidentiality restrictions that govern production use. Annotation is consequently not a separate preliminary step but part of the engineering problem: the system, dataset, and evaluation protocol are all designed around limited access to sensitive documents and the cost of expert review.

	C1	C2	C3
Uses LLM	no	yes	yes
Deterministic	yes	no	no
Fine-tuning	none	none	none
On-premise	yes	yes	yes
Per-fact rationale	no	yes	yes
Agreement signal	no	yes	yes
Uses gold examples	no	no	yes (LOO)
Marginal cost/fact	$\sim$0	medium	medium

Table 4: Operating profiles of the three FINSTRUCT modes. C1 is deterministic and rule-based; C2 and C3 share the two-stage LLM cascade, and C3 additionally injects leave-one-filing-out (LOO) demonstrations into Stage 2.

C Evaluation set and scoring details

From facts to the evaluation set. Not every tagged value in a filing is a well-posed target for logical structure recovery. Continuation tails do not represent independent rendered positions, and hidden facts have no well-defined logical reading position; both are therefore excluded from the 1,493 anchor-eligible facts. Per-section and per-archetype breakdowns of the evaluation set are given in Table 3 and Table 6.

Annotation rationale. Gold annotation requires judgement about how UK statutory accounts are laid out and read—the same expertise the system aims to support—so both annotators are domain experts, and annotation was performed manually from the rendered filings to keep the evaluation fully separated from the systems being evaluated. Per-field inter-annotator agreement is reported in Appendix D.

Annotation conventions. Two conventions affect scoring. First, `sub_heading` is annotated broadly, capturing an intermediate heading that scopes a fact within a section or note rather than only the nearest visible heading. Second, the surrounding-context fields `text_before` and `text_after` use explicit boundary sentinels when no surrounding text exists, so that the absence of context is a labelled outcome rather than an empty string to be guessed.

Matching tiers and outcomes. Exact-match normalisation folds case, whitespace, and punctuation. For each fact and field, the comparison yields one of five outcomes: exact match, fuzzy match, sentinel match, mismatch, or missing value; exact, fuzzy, and sentinel matches are counted as correct. Missing predictions count as incorrect unless the gold label explicitly encodes absence through the relevant sentinel. Denominators exclude fact-field pairs where both gold and prediction are absent. All accuracies are reported with 95% percentile bootstrap confidence intervals from 1,000 resamples over facts (`seed=42`).

D Inter-annotator agreement

Both annotators independently labelled all nine structural fields for all 1,493 anchor-eligible facts, working from the rendered filings without access to each other's labels or to any system output. Table 5 reports per-field percent agreement and Cohen's κ, computed under normalised exact match. This is a stricter criterion than the system scoring in Section 3.2, which additionally admits fuzzy and sentinel matches, so the figures below are a conservative lower bound on agreement.

Agreement is high across the structural fields and lowest on the two surrounding-context fields, where annotators must judge where the relevant context begins and ends. The comparison with C3 is informative: on `text_before` and `text_after` the two annotators agree above 94% while the system reaches 28.5% and 31.8%, indicating that these fields are not intrinsically ambiguous to expert readers and that the gap is a system limitation rather than a ceiling imposed by the task. The same holds, less starkly, for `sub_heading`. Residual disagreements were resolved by the senior annotator to produce the final gold set.

E The deterministic baseline in detail

C1 is the rule-based extractor developed for the deployment rather than a comparator reimplemented for this paper. It runs three phases. Phase 1 builds document structure: it splits the rendered filing into page

Field	% agree	κ	C3
section_title	99.7	0.996	99.7
note_label	98.6	0.982	93.3
sub_heading	98.5	0.977	72.7
row_label	95.8	0.956	85.2
col_header	97.9	0.976	82.1
year	99.2	0.987	84.0
currency_label	99.9	0.998	96.1
text_before	95.0	0.776	28.5
text_after	94.4	0.750	31.8
Macro	97.7	0.933	74.8
Micro	97.7	0.971	85.1

Table 5: Inter-annotator agreement between the two annotators on all 1,493 facts, with C3 accuracy repeated from Table 2 for comparison. Agreement is computed under normalised exact match, a stricter criterion than the system scoring.

cells, detects and classifies section headings, and derives table and column information. Phase 2 assigns each fact one of the five archetypes (Appendix G). Phase 3 dispatches an archetype-specific extractor that returns a single structured record per fact.

The nine fields divide between phases. Heading classification in Phase 1 yields `section_title` alone. Phase 3 produces the other eight: `note_label` and `sub_heading` are assembled from the note number and title of the enclosing note, while `row_label`, `col_header`, `year`, `currency_label`, `text_before` and `text_after` are read directly from the fact's table and text neighbourhood.

Headings are located structurally rather than semantically: candidates are centred container elements carrying a style class and appearing before the first content table on a page. Recurring furniture is then filtered out—the company name, which repeats on almost every page, along with registered-number lines, period subtitles and page numbers—and the surviving candidates are matched against an ordered, first-match-wins pattern list, defaulting to an unknown label.

This design is what F5 defeats. Section identity is recovered from container shape, so a page whose markup does not expose heading containers in the expected form yields no candidates and is skipped, and its facts carry no section. Where candidates exist but all are filtered as furniture, the fallback assigns the cover-page label rather than inheriting the section context of preceding pages, so mid-document pages bearing only recurring headers are mislabelled. The first path is what produces the F5 cohort: with no section resolved, the downstream extractors return nothing and the fact is recorded with all nine fields absent. No exception is raised on either path, which is why the failure is silent and surfaced only under coverage analysis.

F Additional analyses

Context fields under emission and abstention. Accuracy on `text_before` and `text_after` is dominated by absence: 1,305 of 1,493 gold values are absent for `text_before` and 1,302 for `text_after`. Splitting emission from abstention shows that C2 and C3 trade precision against recall rather than one dominating. On `text_before`, emission recall is 72.3 (C2) versus 56.4 (C3) and emission precision 54.2 (C2) versus 66.2 (C3); `text_after` shows the same pattern (recall 66.0 versus 59.2, precision 51.9 versus 63.8). C3 abstains more often (89.3% versus 83.2% on `text_before`). This explains the apparent C2 advantage under accuracy in Table 2: the two modes sit at different points on the same trade-off, and a single accuracy figure over an absence-dominated field obscures that.

Repeated sampling. To separate the contribution of repeated sampling from that of in-context demonstrations, we rescore the 8,958 persisted Stage 2 samples. Plurality voting over $N{=}3$ gains $+0.3$ points macro over the mean single sample, and C3 exceeds C2 at $N{=}1$ (74.6 versus 70.8 macro). The in-context gain is therefore not an artefact of voting.

Rule–LLM hybrid. `year` is the only field where C1 exceeds C3 (85.4 versus 84.0), so a hybrid that routes each field to its stronger mode reduces to rules-for-`year`: macro 75.0 versus 74.8, micro 85.2 versus

85.1. The inference saving is nil, because a single Stage 2 call already emits all eight non-section fields together, so routing individual fields to rules removes no LLM calls. A hybrid that genuinely reduced inference would need a different decomposition—for example, resolving whether a fact is rule-tractable before invoking the model—which we leave to future work.

G Fact archetypes

The structuring frontend assigns each parsed fact one of five archetypes, which route extraction and shape the per-fact input given to every mode.

- **table-numeric**: a numeric cell in a structured financial table. This is the dominant case (1,177 of 1,493 anchor-eligible facts).
- **table-text**: a textual cell in a table, such as a row caption used as a value.
- **narrative-block**: a fact embedded in running prose rather than a table.
- **standalone**: a value outside any table or narrative block, such as a free-standing line item.
- **hidden**: a fact present in the markup but not visibly rendered, for which a logical reading position is undefined. Hidden facts are excluded from the anchor-eligible set and from all evaluation.

The archetype is the backbone of the system: it dispatches the rule-based extractor (Section 2.3) to an archetype-specialised routine, determines the per-fact input representation given to the LLM, and conditions demonstration retrieval in C3. It is also the lens for the per-archetype analysis in Table 6: C1 is competitive only on **table-numeric** facts, while the LLM modes raise accuracy substantially on the three less explicitly encoded archetypes. The **narrative-block** archetype shows the largest improvement from rules to in-context extraction ($21.4\% \rightarrow 84.9\%$, $+63.5$ percentage points), consistent with the comprehension-dependent framing: where structure is carried by prose rather than table geometry, language understanding does most of the work.

H Prompts and schema

Both LLM stages use structured prompts. Stage 1 presents the task definition, the closed inventory of fourteen sections, the target fact, and its rendered neighbourhood, and requests a single section label with a brief rationale. Each of the fourteen sections has its own Stage 2 prompt; all share the skeleton below and differ only in the section-specific field guidance and the layout they expect.

```
[TASK] Recover the logical structure of the
target fact within its section.
[SECTION] {resolved_section_title}
[CONTEXT] {target-centred neighbourhood:
rows, headings, period and currency labels,
surrounding prose}
[TARGET] {fact value + local position}
[FIELDS] note_label, sub_heading, row_label,
col_header, year, currency_label,
text_before, text_after
[OUTPUT] strict JSON: one value per field,
plus a short "rationale" string.
```

A representative Stage 2 prompt body for the balance-sheet section is shown below; the other thirteen follow the same template with section-appropriate field conventions (this example is illustrative of the template, lightly edited for length).

```
You are recovering the logical structure of a
value reported in the BALANCE SHEET of a UK
statutory filing. Using only the context
provided, return the eight remaining fields
for the target value. Columns are reporting
periods; the left-most label column gives the
```

```
line item. If a field has no value in the
filing, return the explicit absence sentinel.
Return strict JSON only.

CONTEXT:
{rendered neighbourhood of the target cell}

TARGET VALUE: {value}

Return: {"note_label": ..., "sub_heading": ...,
"row_label": ..., "col_header": ...,
"year": ..., "currency_label": ...,
"text_before": ..., "text_after": ...,
"rationale": ...}
```

Stage 2 must return a strict JSON object containing the eight field values and a short rationale. Schema validation is part of the system: malformed JSON, missing fields, or values outside the expected type constraints are detected automatically and can be resampled or logged as extraction errors. The rationale is used as a reviewer-facing explanation; we do not claim or evaluate rationale faithfulness.

I Detailed results

This appendix reports the remaining breakdowns: per-archetype accuracy (Table 6), the full pairwise McNemar matrix with Holm–Bonferroni correction (Table 7), and the full per-field results for the second model (Table 8). Two patterns are worth noting. First, `DPL` is the only section where the zero-shot LLM regresses relative to the rule-based extractor ($93.6 \rightarrow 82.6$), which in-context demonstrations then recover (95.2)—a section-level echo of the `year` field result. Second, the per-archetype and per-section breakdowns localise where the LLM modes contribute most: the least explicitly encoded archetypes and the narrative report sections.

Archetype	n (facts)	C1	C2	C3
table-numeric	1,177	69.1	82.0	85.9
table-text	118	39.8	68.9	76.1
narrative-block	118	21.4	68.4	84.9
standalone	80	32.4	75.2	76.4

Table 6: Per-archetype accuracy (%) on the Qwen3-14B-AWQ run, anchor-eligible facts only (HIDDEN excluded by construction). Cells show micro accuracy: total correct cells / total non-both-missing cells across the nine prediction fields, within each archetype. n is the fact count per archetype (sums to 1,493). Largest C1→C3 gain is on *narrative-block* (+63.5 pp), where the rule-based extractor's text-window heuristics are weakest and the LLM cascade does the heaviest lifting; *table-numeric* (the bulk of the corpus) shows the smallest delta because C1 already exploits column / row geometry.

Field	C1 vs C2	C1 vs C3	C2 vs C3
`section_title`	6.0e-52 ✓	6.0e-52 ✓	1.00 ×
`note_label`	3.9e-60 ✓	4.9e-76 ✓	3.6e-2 ✓
`sub_heading`	3.7e-97 ✓	2.6e-148 ✓	9.1e-20 ✓
`row_label`	7.4e-57 ✓	1.1e-60 ✓	1.00 ×
`col_header`	2.4e-52 ✓	1.5e-50 ✓	1.00 ×
`year`	0.295 ×	4.9e-6 ✓	9.7e-12 ✓
`currency_label`	5.8e-44 ✓	3.3e-41 ✓	0.500 ×
`text_before`	2.5e-21 ✓	1.6e-10 ✓	4.5e-4 ✓
`text_after`	4.8e-17 ✓	5.1e-17 ✓	0.435 ×
sig. count	8 / 9	9 / 9	4 / 9

Table 7: Pairwise McNemar exact test, Holm-Bonferroni adjusted across the family of 9 fields × 3 pairwise comparisons = 27 hypotheses. Cells show the Holm-adjusted p-value; ✓ marks $p < 0.05$ and × marks n.s. The C1-vs-C2 column reproduces the raw McNemar exact test stored in `summary.json` to 6 significant figures before correction; raw values are available in the JSON dump and the run log. The only pair where raw vs Holm matters substantively is `year` C1-vs-C2 (raw 0.0491, Holm 0.295) — the body's characterisation of `year` as n.s. for C1-vs-C2 reflects the Holm-corrected verdict.

Field	C1 [lo, hi]	C2 [lo, hi]	C3 [lo, hi]
section_title	86.5 [84.7, 88.1]	95.8 [94.8, 96.8]	99.7 [99.4, 99.9]
note_label	60.0 [56.9, 63.9]	62.8 [59.6, 66.2]	79.7 [76.7, 82.5]
sub_heading	3.4 [2.2, 4.9]	37.3 [34.0, 40.9]	58.5 [54.7, 61.9]
row_label	65.2 [62.3, 68.0]	59.0 [56.2, 61.6]	73.2 [70.8, 75.7]
col_header	60.9 [58.1, 63.7]	47.8 [45.0, 51.1]	63.1 [60.2, 65.8]
year	85.4 [82.9, 87.8]	43.4 [40.6, 46.3]	56.7 [53.5, 59.9]
currency_label	82.8 [80.6, 84.8]	90.9 [89.3, 92.5]	91.4 [89.6, 93.0]
text_before	6.1 [3.8, 9.2]	11.5 [7.9, 15.1]	18.5 [14.4, 22.8]
text_after	5.3 [2.8, 8.1]	2.8 [1.2, 4.8]	10.1 [7.0, 14.0]
Macro	50.6 [49.4, 51.9]	50.1 [49.1, 51.1]	61.2 [60.0, 62.6]
Micro	63.1 [61.3, 64.8]	61.7 [60.5, 62.9]	72.6 [71.3, 73.9]

Table 8: Per-field accuracy (%) on Llama-3.1-8B-Instruct-AWQ-INT4 (the second open-weight instruction-tuned model in the cross-model robustness check), with 95% percentile bootstrap CIs (1,000 resamples, seed=42, fact-level resampling). C1 is the rule-based extractor and is identical across runs by construction. C3 $>$ C2 holds on every field; absolute level is materially below Qwen, particularly on the structural fields (`year`, `col_header`, `row_label`, `note_label`) where the smaller model under-uses the page-rendered structural context.

J Within-family capacity sweep

To test whether the in-context effect is specific to the 14B model, we repeat the C2 and C3 experiments across the Qwen3 family at four sizes (Table 9). All sizes use the same inference configuration as the primary run (Section 3.3): max model length 32,768, Stage 1 decoded greedily (T=0, 64 tokens), Stage 2 sampled with N=3 at T=0.7 (top-p 0.9, 1024 tokens), and C3 using leave-one-filing-out k=3 demonstrations. C1 is model-agnostic (50.6 macro / 63.1 micro) and unchanged across rows.

Model	Quant.	C2	C3	Δ
Qwen3-1.7B	BF16[†]	42.1 / 50.0	52.2 / 62.2	+10.1
Qwen3-4B	AWQ	53.8 / 64.2	63.8 / 72.3	+10.0
Qwen3-8B	AWQ	61.7 / 70.8	69.1 / 78.3	+7.4
Qwen3-14B	AWQ	71.0 / 80.4	74.8 / 85.1	+3.8

Table 9: Capacity sweep: macro / micro accuracy (%) by model size, with the macro C3−C2 gain (Δ). C3 $>$ C2 at every size, but the in-context gain shrinks as capacity grows. [†]No official Qwen3-1.7B AWQ build exists, so the 1.7B tier is run at native BF16—a full-precision upper bound, not directly comparable to the AWQ INT4 tiers; we read its point conservatively.

Field	1.7B	4B	8B	14B
section_title	83.1	92.8	95.7	99.7
note_label	45.1/70.6	58.3/79.0	79.5/87.6	86.6/93.3
sub_heading	18.4/38.4	29.7/53.3	44.8/60.6	55.2/72.7
row_label	58.3/65.7	68.0/76.4	83.5/83.2	82.1/85.2
col_header	43.3/48.7	61.9/71.0	69.5/73.9	80.6/82.1
year	40.0/49.9	58.4/69.3	64.9/81.7	73.9/84.0
currency_label	80.5/89.0	91.9/92.9	92.1/94.6	95.8/96.1
text_before	6.4/13.7	12.0/19.3	14.7/24.6	34.7/28.5
text_after	4.0/10.7	11.7/19.8	11.0/20.1	30.8/31.8

Table 10: Per-field accuracy (%) across model sizes, shown as C2/C3. `section_title` is identical for C2 and C3 by construction (C3 reuses C2's Stage-1 decision), so a single value is shown.

Findings. Accuracy scales monotonically with capacity for both modes (C2 macro 42.1 $\rightarrow$ 71.0; C3 52.2 $\rightarrow$ 74.8 across 1.7B$\rightarrow$14B), and C3 exceeds C2 at every size, confirming that the in-context effect is not an artefact of one model. The size of that effect, however, shrinks with capacity—from +10.1 points at 1.7B to +3.8 at 14B—so demonstrations help weaker models most. At 1.7B, zero-shot extraction falls below the rule-based baseline (C2 42.1 $<$ C1 50.6) and only in-context extraction recovers above it. This mirrors the second-model result (Section 4.5): the weaker Llama-3.1-8B shows a larger in-context gain

than the larger Qwen3 models, indicating that base capacity, not family, governs how much demonstrations contribute.

K API-equivalent cost estimate

The binding reason for on-premise inference is confidentiality: client filings cannot be sent to a commercial API, so the hosted option is unavailable regardless of price. For completeness, Table 11 estimates the inference spend the API option would have incurred at mainstream rates at the time of writing. The figures are illustrative: they assume an input-dominated token split, apply no batch or caching discounts, and reflect a pricing snapshot that may change. Even on conservative assumptions the recurring cost is non-trivial and, unlike the fixed cost of owned hardware, scales linearly with the firm's filing volume.

Inference option	**C2**	**C3**
	(USD / 1k facts)	
Commercial API, model A	42	63
Commercial API, model B	47	70
Commercial API, model C	34	50
FINSTRUCT (on-premise GPU)	$\sim$0	$\sim$0

Table 11: Illustrative API-equivalent inference cost per 1,000 facts, estimated from observed token counts and a pricing snapshot at the time of writing, under an input-dominated assumption and without batch or caching discounts. The comparison is illustrative only: confidentiality makes hosted API inference unavailable in our deployment setting, and prices may change over time.