Beyond Isolated Dots: Benchmarking Structured Table Construction as Deep Knowledge Extraction

Anonymous ACL submission

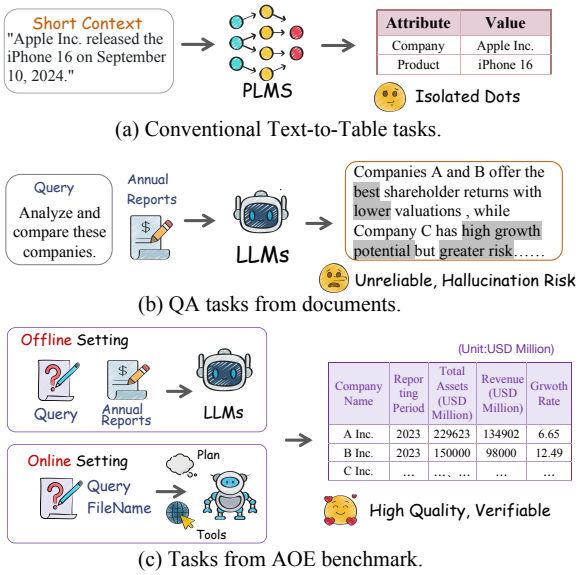
Abstract

With the emergence of large language models (LLMs), there is an expectation that they can effectively extract structured information from complex, real-world documents. However, most LLMs generate paragraph-style answers that are chaotic, disorganized, and untraceable. To bridge this gap, we introduce the Arranged and Organized Extraction Benchmark (AOE), a new bilingual benchmark featuring long-form documents (ranging from 64k to 1.8M tokens per task) designed to systematically evaluate the ability of LLMs to comprehend fragmented information and reconstruct it into an organized table. Unlike conventional Text-to-table tasks, which rely on narrow domains and artificially constructed inputs, AOE includes 12 carefully designed tasks across four diverse domains, where each task schema is grounded in real-world analytical queries—columns directly correspond to the analytical objective, and schema guidance is provided to models to isolate extraction ability from formatting variance. In our experiments, we evaluated 12 state-of-the-art LLMs across open-weight and proprietary categories. Even the most advanced models struggle significantly, with the top-performing LLM achieving a Cell F1 score of only 42.7%, demonstrating a widespread **"Area of Effect" (AOE)** impact on all tested systems. The benchmark is available at `https://anonymous.4open.science/r/AOE-Benchmark/`.

1 Introduction

With the emergence of large language models (LLMs), there is a growing expectation that they can effectively extract and synthesize information from complex real-world documents such as financial reports, legal cases, and academic papers (Wang et al., 2024b; Bai et al., 2023; Zhang et al., 2024a; Li et al., 2025).

Conventional text-to-table tasks, often exemplified by simplified scenarios such as those in Figure 1(a) (e.g., short single-document inputs (Wu et al., 2022; Sundar et al., 2024) and narrow domain coverage (Wang et al., 2024a)), fall short of addressing this complexity. While LLMs offer promise, they often generate chaotic, untraceable paragraph-style answers (conceptually illustrated in Figure 1(b)), severely lacking in both usability and trustworthiness.



(a) Conventional Text-to-Table tasks.

(b) QA tasks from documents.

(c) Tasks from AOE benchmark.

Figure 1: Differences between AOE and previous tasks. Unlike (a) extracting "Isolated Dots" or (b) generating "Unverifiable Output," (c) AOE requires constructing "Organized Evidence" for verifiable analysis.

Existing benchmarks leave a critical gap. Long-context benchmarks like Loong (Wang et al., 2024b) and L-EVAL (An et al., 2024) focus on comprehension and question answering without requiring structured outputs. Context learning benchmarks (Dou et al., 2026b,a) evaluate whether models can acquire knowledge from context, but operate on single documents. Text-to-table benchmarks (Wang et al., 2024a; Deng et al., 2024; Jiang et al., 2024) require structured outputs but are limited to single-document inputs with short

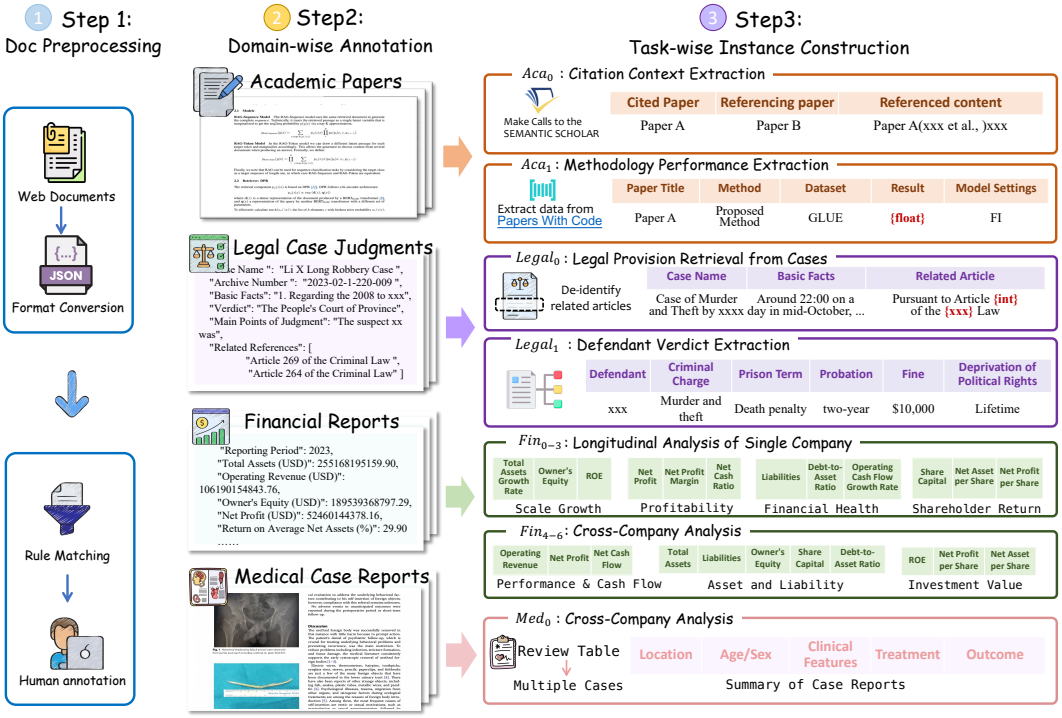


Figure 2: Construction process of our AOE Benchmark.

contexts. No existing benchmark jointly evaluates multi-document reasoning and structured knowledge construction over long contexts.

To bridge this gap, we introduce **AOE (Arranged and Organized Extraction)**, a bilingual benchmark spanning four diverse domains (academic, financial, legal, and medical) with an average input context of 171k tokens across 2,166 source documents, as illustrated in Figure 1(c). AOE is the first benchmark to jointly evaluate long-context multi-document comprehension and structured table construction. Its core features include:

Naturally Grounded Multi-Document Tasks: Unlike benchmarks that artificially concatenate documents to manufacture long contexts, every AOE task originates from a genuine analytical need. Ground-truth tables are derived from real-world artifacts—medical literature review tables summarizing clinical cases across publications, academic leaderboards aggregating experimental results across papers, financial comparative analyses spanning multi-year reports, and legal verdict summaries across related cases. Each task requires synthesizing information from multiple documents (up to 13 per instance) into structured tables, with contexts ranging from 64k to 1.8M tokens.

Hierarchical Evaluation Framework: Our automated pipeline evaluates each output at three levels: (1) structural parsability, (2) overall quality via an LLM-evaluator, and (3) cell-level factual accuracy using F1 scores.

Dual Evaluation Settings: We assess models in a **document-grounded (Offline)** setting to test core synthesis abilities, and an **open-world agentic (Online)** setting for evaluating end-to-end information seeking and structuring pipelines.

Even state-of-the-art models struggle significantly, with the top-performing LLM achieving only 48.8% *Cell F1*, demonstrating a widespread **"Area of Effect" (AOE)** impact on all tested systems.

2 AOE Benchmark

Our benchmark evaluates LLMs on multi-step extraction and inferential reasoning tasks involving implicit calculations across Academic, Financial, Legal, and Medical domains, with statistical summaries and task definitions detailed in Table 2 and Table 16, respectively.

Existing benchmarks for evaluating LLMs' structured information extraction exhibit critical limitations. Long-context QA benchmarks (Wang et al., 2024b; An et al., 2024) focus on comprehension without structured outputs, while Text-to-table tasks are constrained by single-document inputs (Wang et al., 2024a; Deng et al., 2024; Jiang

Benchmark	Languages	Task Type	Domain	#Docs	Instances	Multi-Doc	Context	Structured Output
Long-Context Understanding & Context Learning								
L-EVAL (An et al., 2024)	EN, ZH	Long-Context Understanding	19	508	2,000+	✗	3k–200k	✗
Loong (Wang et al., 2024b)	EN, ZH	Multi-Doc Long-Context QA	3	1,600	1600	✓	110K	✗
LongBench v2 (Bai et al., 2025)	EN	Long-Context Multitask QA	∼15+	–	503	✓	104k	✗
CL-bench (Dou et al., 2026b)	EN	Context Learning	4	500	1,899	✗	10.4k	✗
Text-to-Table & Structured Construction								
LiveSum (Deng et al., 2024)	EN	Summarization to Tables	1	1,122	1,122	✗	1.3k	✓
CPL (Jiang et al., 2024)	ZH	Legal Text-to-Table w/ KG-Aug	1	200	200	✗	1.1k	✓
DocTabQA (Wang et al., 2024a)	EN	Long-Doc to Table	1	828	828	✗	19.4k	✓
AOE (Ours)	**EN, ZH**	**Multi-Doc Struct KC**	**4**	**2,166**	**447**	✓	**171k**	✓

Table 1: Comparison of AOE with existing benchmarks in Long-Context Understanding and Structured Construction. KC: Knowledge Construction. AOE uniquely combines long multi-document context with structured output generation across 4 domains.

Metric	Academic	Financial	Legal	Medical
Basic Info				
Language	EN	ZH, EN	ZH	EN
Tables (#)	74	224	75	74
Document-Level				
Documents (#)	257	944	625	340
Avg Tokens	69k	437k	7k	22.5k
Table-Level				
Rows (avg/max)	4.0/15	4.2/5	11.6/55	4.6/5
Columns (avg/max)	4.0/4	7.4/9	5.9/10	10.9/17
Docs (avg/max)	3.5/5	4.2/5	8.3/10	4.6/5
Cells (avg/max)	15.9/60	30.8/36	78.5/550	50.6/85
Total Input Tokens	250K	1.8M	64K	104K

Table 2: Data statistics of AOE benchmark.

et al., 2024), fixed schemas (Deng et al., 2024; Jiang et al., 2024), or synthetic data (Wu et al., 2022; Hsieh et al., 2024; gkamradt, 2025). Agent-focused benchmarks like GAIA (Mialon et al., 2024) and FRAMES (Krishna et al., 2025) support multi-document reasoning but lack evaluation of structured knowledge construction.

To bridge these gaps, AOE constructs a challenging bilingual benchmark across Academic, Financial, Legal, and Medical domains (Table 2).

Input lengths vary substantially across domains: Financial tasks average 1.8M tokens, Academic tasks 250K tokens, Medical tasks 111K tokens, and Legal tasks 64K tokens. For models with bounded context windows, inputs exceeding the maximum length are truncated to fit. Importantly, this truncation does not imply that the discarded suffix is irrelevant: as shown in Appendix **??**, a non-negligible fraction of ground truth answers (5.6% overall, and up to 9.9% in the Financial domain) reside beyond the 256k-token boundary. This highlights a fundamental limitation of current long-context models: even when provided with massive context windows, critical information may still lie out of reach due to hard truncation.

2.1 Data Collection

We curated the AOE dataset exclusively from publicly available, real-world long-form documents across four domains. Critically, unlike benchmarks that artificially concatenate unrelated documents to create long contexts, every AOE instance reflects a genuine multi-document analytical need: ground-truth tables are derived from real-world artifacts such as academic leaderboards, financial comparative reports, legal verdict summaries, and medical literature review tables. This ensures that the benchmark simultaneously tests practical utility and core model capabilities—long-context comprehension, cross-document reasoning, and structured output generation.

Academic: English research papers sourced from Semantic Scholar and Papers With Code. Ground-truth tables correspond to dataset leaderboards that aggregate experimental results across multiple papers.

Financial: Bilingual (Chinese and English) A-share company annual reports (2020–2023) sourced from CNINFO. Tasks mirror real analyst workflows requiring longitudinal and cross-company comparisons.

Legal: Chinese civil law judgments and statutory provisions collected from the People's Court Case Library. Tables capture structured verdict summaries and statutory provision mappings across related cases.

Medical: English biomedical case report papers from PubMed Central (PMC) Open Access Subset[1]. We search for systematic reviews in Q1/Q2 journals that contain "summary of reported cases"

[1]https://pmc.ncbi.nlm.nih.gov/

Example: Constructing a Medical Instance

1. Review paper discovered via PubMed: Li et al. (2024b) contains Table 1: summary of cases of NSCLC-to-SCLC transformation (2017–2023).
2. Ground-truth table extracted (12 columns):
source_doc, Age, Sex, Country, Smoking_status, Mutational_status, Medication_before, Medication_after, CNS_metastasis, OS_after_transformation, OS
Each row cites one independent case report; values curated by the review authors.
3. Source papers collected: For each row, retrieve the cited OA case report. Filter: must be Open Access and describe exactly one case. Result: 7 of 12 rows pass → 7 input documents.
4. Human verification: Confirm cell values are traceable to source papers, no duplicates, columns consistent.

Table 3: Medical data construction pipeline (real instance from our benchmark). The ground-truth table originates from a peer-reviewed systematic review in a Q1 journal, ensuring annotation quality without additional crowd-sourcing.

tables—structured tables where each row cites one independent case report and columns represent clinical attributes. These review tables serve as authoritative ground truth (curated by domain-expert authors and peer-reviewed). We then retrieve the original Open Access case report for each cited row as input documents. Table 3 shows a concrete example; full details are in Appendix D.

All documents were pre-processed from PDF to markdown.

2.2 Task Formulation

AOE defines 12 tasks across four domains, each requiring models to construct structured tables from multiple unstructured documents. All tasks share three core challenges:

Cross-Document Synthesis. Each task instance takes multiple source documents $D = \{d_1, \ldots, d_M\}$ as input (up to 13 per instance). Models must locate, align, and integrate information scattered across independent documents with heterogeneous formats and terminologies.

Task-Dependent Schema. Each task defines a schema that is tightly coupled with the analytical objective of the query—columns directly reflect the real information needs (e.g., financial growth indicators, legal verdict components, clinical attributes). To ensure fair evaluation that isolates extraction quality from schema formatting, we provide explicit schema guidance in the evaluation prompt. Models must still interpret the query semantics to populate the table correctly, but are not expected to freely invent column structures.

Multi-Step Reasoning. Beyond surface-level extraction, tasks require implicit calculations (e.g., year-over-year growth rates in financial reports), inferential reasoning (e.g., mapping case facts to statutory provisions), and cross-reference resolution (e.g., aligning clinical data reported in different formats across papers).

The four domains—Academic, Financial, Legal, and Medical—are designed to cover a broad spectrum of reasoning demands, including citation analysis, longitudinal and cross-entity financial comparison, legal provision retrieval and verdict extraction, and clinical evidence synthesis. Detailed per-task definitions are provided in Appendix F.

2.3 Ground-Truth Annotation Process

To ensure reproducibility and high quality, we implemented a rigorous two-phase annotation process involving domain experts.

Phase 1: Expert-Led Schema Design. This phase was led by domain experts (practitioners in finance and law). Experts defined meaningful queries simulating practical analytical needs and constructed structured table schemas (defining columns and data types) grounded in the query's analytical objective. The resulting schema serves as the gold standard for evaluation. At inference time, models receive the query along with schema guidance specifying the expected columns, ensuring that evaluation focuses on extraction and reasoning quality rather than schema formatting (see Section 2.2).

Phase 2: Independent Annotation and Adjudication. The core annotation team consisted of graduate students with relevant domain expertise. Each task instance was assigned to two annotators who worked independently. The resulting tables were automatically compared, and discrepancies were escalated to senior domain experts for final adjudication. Annotators do not read documents linearly but leverage search and navigation tools to locate relevant information efficiently, completing tables in approximately 15–25 minutes each. Concrete examples of annotation guidelines and adjudicated tables are provided in Appendix E.

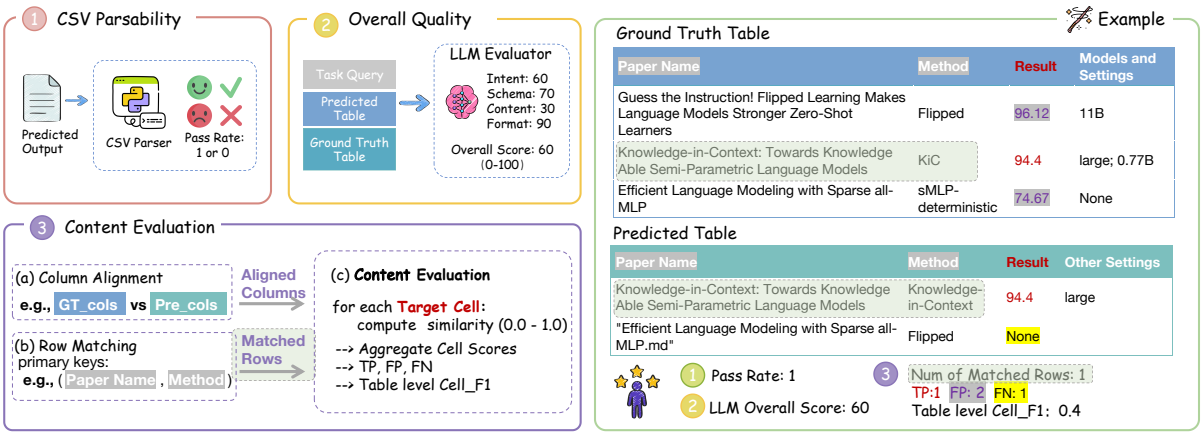


Figure 3: Overview of the Automated Evaluation Pipeline. The pipeline evaluates generated tables from three perspectives: (1) CSV Parsability for basic structural correctness; (2) Overall Quality via an LLM evaluator for a holistic score; and (3) Content Evaluation to calculate a Cell F1 score based on cell-level comparison after column and row alignment.

Annotation Quality. To verify data quality, we conducted a blind re-annotation study on 436 cells across all four domains. The initial Exact Match rate was 98%, and after expert adjudication of discrepancies, the final agreement reached 100%, confirming that the benchmark consists of objective, verifiable facts. A human performance study is reported in Section 3.5.

2.4 Benchmark Evaluation

2.4.1 Task Definition

Given a set of source documents $D = \{d_1, \ldots, d_M\}$ and a natural language query q describing the analytical goal, the task is to produce a structured table $T = (C, R)$, where $C = \{c_1, \ldots, c_K\}$ is the set of columns defined by the task schema, and $R = \{r_1, \ldots, r_N\}$ is a set of rows, with each cell v_{ij} containing information extracted or inferred from D. The schema C is provided to the model as part of the evaluation prompt (schema guidance); the number of rows N must be determined from the document content.

2.4.2 Automated Evaluation Pipeline and Metrics

AOE is designed with a hierarchical evaluation pipeline that examines generated tables from basic parsability to fine-grained cell-level accuracy. We report three primary metrics: (1) *LLM Score (Overall Quality)*, (2) *Row F1 (Row Retrieval)*, and (3) *Cell F1 (Content Evaluation)*. The pipeline is illustrated in Figure 3.

The process begins by mapping model-generated tables to the ground-truth schema—a necessary preprocessing step to enable fine-grained evaluation. We then evaluate table quality using the three core metrics described below (see Appendix H.2 for all prompt details).

LLM Score (Overall Quality) We prompt a strong LLM evaluator to rate the overall table on a 0–100 scale across four dimensions: Intent Understanding, Schema Construction, Content Accuracy, and Format Compliance. The final score is computed as a weighted average of these dimensions with weights of 0.2, 0.3, 0.3, and 0.2, respectively.

Row F1 (Row Retrieval) After aligning columns via an LLM-based mapper (yielding matched columns C_{aligned}.), we identify correctly predicted rows by matching against ground-truth primary keys. We match rows between predicted and ground truth tables using pre-defined primary keys, obtaining a set of matched rows R_{matched}. Row F1 is computed as the harmonic mean of row-level precision and recall.

Cell F1 (Content Evaluation) For each aligned cell pair at position (i, j) where $i \in R_{\text{aligned}}$ and $j \in C_{\text{matched}}$, with predicted content v_{ij}^{pred} and gold content v_{ij}^{gold}, an LLM evaluator rates semantic similarity on a continuous 0-1 scale, producing a rating score $s_{ij} \in [0, 1]$. This soft-matching approach allows partial credit for semantically similar cells (see Appendix H.2.3 for detailed rating criteria).

This comprehensive evaluation suite combines automated structural validation, fine-grained content accuracy measurement, and holistic quality assessment.

Model	Pass Rate	Academic			Legal			Fin-S		Fin-L		Medical			Avg. Cell F1
		LLM Score	Row F1	Cell F1	LLM Score	Row F1	Cell F1	Row F1	Cell F1	Row F1	Cell F1	LLM Score	Row F1	Cell F1	
Open-Weight Models (sorted by Avg. Cell F1 ↑)															
DeepSeek-V3.2	85.9	80.1	44.4	36.9	79.1	77.1	45.5	96.0	57.1	81.6	30.1	65.2	88.1	23.5	38.6
Qwen-3.6-Plus	90.2	81.8	67.6	52.2	82.7	77.7	48.2	90.1	48.8	68.9	22.7	67.2	93.1	27.8	39.9
Kimi-K2.6	96.0	79.0	47.5	34.5	81.3	78.5	43.3	92.1	**63.4**	65.6	35.3	62.9	96.4	24.4	40.2
GLM-5.1	91.7	81.7	52.9	42.1	79.2	79.1	44.4	94.1	52.9	78.9	33.4	62.3	92.8	31.2	40.8
Kimi-K2.5	94.2	79.1	47.8	34.1	79.5	**85.5**	50.0	94.1	62.5	67.3	35.0	64.9	91.4	29.3	42.2
DeepSeek-V4-Pro	96.6	**82.8**	**77.5**	**58.6**	76.3	80.4	**50.4**	95.9	52.8	74.8	24.6	61.3	94.3	26.9	**42.7**
Proprietary Models (sorted by Avg. Cell F1 ↑)															
Gemini-3.1-Pro	92.4	79.2	56.8	42.8	79.7	70.3	44.3	95.0	53.1	78.6	27.1	53.1	81.3	23.0	38.1
Claude-Sonnet-4.6	93.5	79.2	56.2	40.2	84.4	67.9	36.6	85.1	60.4	71.0	32.7	**67.3**	90.6	28.8	39.7
Claude-Opus-4.6	**99.8**	79.8	57.1	36.3	79.9	73.1	39.0	93.1	56.0	**87.4**	38.6	65.4	92.8	**31.8**	40.3
Gemini-3-Flash	82.3	80.4	61.4	44.2	80.2	71.3	38.3	69.3	44.4	87.0	**55.1**	48.0	68.2	20.4	40.5
GPT-5.4	89.5	79.4	63.8	48.2	**85.0**	71.0	40.1	**97.0**	59.9	82.6	32.7	63.7	93.5	23.7	40.9
Seed-2.0-Pro	87.1	81.6	63.1	45.3	83.5	71.4	40.0	90.8	60.1	71.7	40.7	65.4	**94.9**	26.0	42.4

Table 4: Main results on AOE benchmark with Financial domain split into **Fin-S** (short, ≤200k total tokens) and **Fin-L** (long, >200k total tokens). Models are sorted by average Cell F1 (ascending) within each category. Instances where the model fails to produce valid CSV are counted as Cell F1 = 0 in the denominator, ensuring Pass Rate is reflected in final scores. All results are from run_0. **Bold** values indicate the top score in each column. All values in %.

Evaluator Robustness. To verify the reliability of our LLM-based scoring pipeline, we conducted stability analyses across three dimensions: (1) *Inter-judge consistency*: two independent LLM judges evaluated 6,866 cells, yielding a Pearson correlation of 0.9134 ($p < 0.001$); (2) *Intra-judge consistency*: repeated scoring (8 runs) on 5,636 cells showed variance ≤ 0.03 across all value types; (3) *Ranking robustness*: bootstrap analysis confirmed model rankings shift by less than 1 position on average. Full details are provided in Appendix J.

3 Experiments

3.1 Experimental Setup

Models Under Evaluation. We evaluate twelve frontier LLMs spanning two categories. **Open-weight models** include the latest releases from leading open-source efforts: DeepSeek-V3.2 (DeepSeek-AI et al., 2025) and DeepSeek-V4-Pro (DeepSeek-AI, 2026), Qwen-3.6-Plus (Qwen Team, 2026), GLM-5.1 (Z.ai, 2025), and Kimi-K2.5/K2.6 (Moonshot AI, 2026a,b). **Proprietary models** represent the flagship offerings from major providers: GPT-5.4 (OpenAI, 2026), Claude-Opus-4.6 and Claude-Sonnet-4.6 (Anthropic, 2026), Gemini-3.1-Pro and Gemini-3-Flash (Google DeepMind, 2026), and Seed-2.0-Pro (ByteDance Seed Team, 2026). All evaluations use GPT-4o (OpenAI, 2024) as the LLM judge for cell-level scoring.

Generation Settings. Our primary evaluation focuses on the **Direct** setting, where full source documents are provided as input for single-pass inference to assess native long-context reasoning. To further investigate model capabilities under varying degrees of autonomy, we introduce three additional variants: **RAG** (chunk-based retrieval), **Offline Agent** (iterative local document navigation), and **Online Agent** (open-web search integration). The detailed definitions, evaluated capabilities, and comparative analysis of these settings are thoroughly discussed in Section 3.3. Implementation details and tool references are provided in Appendix C.

3.2 Main Results: The Bottleneck of Long Contexts

Table 4 shows that *no model has solved the AOE challenge*: the best performer (DeepSeek-V4-Pro) achieves only 42.7% Cell F1, leaving ∼57% of target facts missed or incorrect.

The open–proprietary gap is narrowing. Open-weight models (DeepSeek-V4-Pro, Qwen-3.6-Plus, GLM-5.1) now match or exceed several proprietary counterparts, suggesting that long-context structured reasoning is no longer gated by model access.

Pass Rate is Gemini's bottleneck. Despite strong extraction quality when outputs are valid, Gemini-3.1-Pro is held back by a comparatively

low Pass Rate. We observe that Gemini-3.1-Pro frequently generates Markdown-formatted tables even when the prompt explicitly requests CSV output, suggesting that its instruction tuning heavily favors Markdown as the default structured output format. Even if the evaluation pipeline will be later set to convert to CSV format during evaluation, some information loss will still occur.

3.3 Generation Settings Comparison

RAG: semantic retrieval is insufficient. Introducing FAISS-based chunk retrieval consistently degrades Cell F1 compared to Direct in Academic and Legal domains (e.g., DeepSeek-V4-Pro Academic: 58.6% → 30.7%; Qwen-3.6-Plus Legal: 55.2% → 30.9%), demonstrating that standard embedding-based semantic similarity cannot capture the multi-hop, cross-document dependencies required by AOE. The sole exception is Financial, where RAG roughly maintains or marginally improves performance for some models (e.g., Gemini-3.1-Pro: 41.6% → 45.9%) by mitigating context truncation on ultra-long inputs (∼1.8M tokens). A detailed length-stratified analysis is provided in Appendix G.

Offline Agent: iterative browsing helps ultra-long documents. On the Financial domain, most models achieve substantial gains through iterative document browsing (Claude-Opus-4.6: 46.7% → 60.9%; DeepSeek-V4-Pro: 39.9% → 62.0%; GPT-5.4: 46.3% → 59.2%), suggesting that selective, tool-guided reading effectively circumvents context window limitations. However, this advantage does not generalize: in Academic and Legal, where cross-document synthesis is essential, the agent's section-level view fragments the global context needed to align entities across documents, leading to overall degradation. A notable outlier is Gemini-3.1-Pro, whose Financial Cell F1 *drops* from 41.6% to 23.8%—opposite to all other models—indicating that effective tool use itself remains an unsolved challenge for certain model families.

Online Agent: web search struggles with structured extraction. Online Agent yields the lowest Cell F1 across nearly all model–domain combinations, with most scores falling below 20%. Although all source documents are publicly accessible, web search engines return page-level results optimized for topical relevance rather than the fine-grained factual passages needed for table construction. The gap between Online and Direct is starkest in Financial (e.g., DeepSeek-V4-Pro: 39.9% → 12.8%) and Academic (e.g., GPT-5.4: 48.2% → 15.2%), where scattered numerical evidence is nearly impossible to recover through keyword-based queries.

Model	Setting	CF1	ColR	ColP	RowR	CSV
Claude-Sonnet-4.6	Prompt-All	19.6	71.5	63.9	92.6	64.9
	Schema-Guided	28.4	93.1	83.9	92.1	95.9
	Δ	**+8.8**	**+21.6**	+20.0	−0.5	+31.0
DeepSeek-V3.2	Prompt-All	20.3	66.9	59.2	95.4	52.7
	Schema-Guided	23.4	89.1	80.1	89.8	90.5
	Δ	**+3.1**	**+22.2**	+20.9	−5.6	+37.8
Qwen-3.6-Plus	Prompt-All	17.2	60.3	53.9	86.9	67.6
	Schema-Guided	27.7	90.7	81.6	91.9	93.2
	Δ	**+10.5**	**+30.4**	+27.7	+5.0	+25.6

Table 5: Schema-Guided vs. Prompt-All on the Medical domain (74 instances). CF1 = Cell F1, ColR = Column Recall, ColP = Column Precision, RowR = Row Recall, CSV = CSV parse rate. All values in %.

3.4 Ablation: Schema Guidance

A key design choice in AOE is whether to provide explicit schema guidance (column definitions) alongside the query, or let models freely determine the output schema. We ablate this choice on the Medical domain, where schema divergence is most severe, comparing **Prompt-All** (free-form schema) against **Schema-Guided** (explicit column definitions provided).

Table 5 reveals that schema guidance is essential for meaningful evaluation. Column Recall increases by 21–30 percentage points across all models, confirming that without guidance, models construct plausible but divergent schemas unmappable to the gold standard. Cell F1 also improves by 3–11 points, suggesting that schema guidance helps models focus extraction on the correct attributes. CSV parse rates increase by 25–38 points, as explicit schema constraints reduce malformed outputs. A detailed case study is provided in Appendix K.

3.5 Human Benchmark

A graduate-level annotator (non-expert in CS and medical domains), permitted to use online search tools, completed a subset of AOE tasks. The annotator substantially outperforms all LLMs on Financial (92.9% vs. top models struggling below 64% across splits) and Legal (85.8% vs. best model 50.4%), but scores comparably on Academic (42.4%) and below models on Medical (34.0%)—

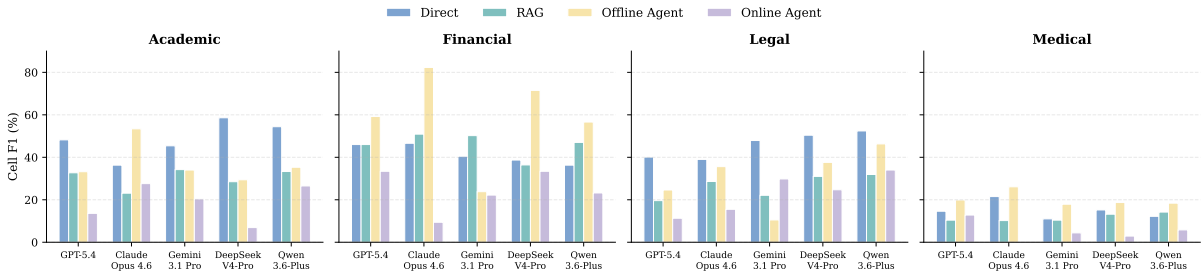


Figure 4: Cell F1 (%) across four generation settings—Direct, RAG, Offline Agent, and Online Agent—for five representative models, broken down by domain.

consistent with domain expertise requirements in these specialized fields. Full results and analysis are in Appendix L.

3.6 Evaluation Framework Reliability

Run-to-Run Stability. Across five independent runs on five models, the mean Cell F1 standard deviation is 0.026, with moderate ranking concordance (Kendall's $W = 0.664$). Instability mainly arises from tight mid-tier score clusters. Full results in Appendix J.

Cross-Judge Consistency. Four independent LLM judges yield pairwise Pearson correlations of $r = 0.71$–0.95 (mean 0.82), with the top-ranked model holding rank #1 under three of four judges. This confirms our evaluation captures genuine quality differences rather than judge-specific biases.

4 Related Work

4.1 Long-Context Understanding & Context Learning

Existing long-context benchmarks often suffer from limitations: synthetic tasks like NIAH (Kamradt, 2023) and Counting Stars (Song et al., 2024) only test surface-level retrieval, while early datasets (Bai et al., 2023; Li et al., 2024a) lack sufficient length. Although recent works (An et al., 2024; Dong et al., 2024b; Qiu et al., 2024; Zhang et al., 2024b) and flexible methods like Ruler (Hsieh et al., 2024) or Babilong (Kuratov et al., 2024) improve comprehensiveness, they primarily focus on passive comprehension. More recently, CL-bench (Dou et al., 2026b) and CL-bench Life (Dou et al., 2026a) evaluate *context learning*—whether models can acquire new knowledge from provided context and apply it to solve tasks. Oolong (Bertsch et al., 2025) specifically targets information aggregation over long inputs up to 128K tokens, requiring models to classify and count scattered evidence. However, these benchmarks operate on single-document inputs and do not require structured output generation.

4.2 Text-to-Table Generation

Early approaches relied on rigid schemas (Dong et al., 2024a) and synthetic data (Wu et al., 2022; Sundar et al., 2024), failing to capture real-world complexity. LLM-based methods have shifted toward formatting (Tang et al., 2024), reconstruction (Singh et al., 2024; Ni et al., 2023), domain-specific extraction (Wang et al., 2024a; Zhao et al., 2024), and intermediate organization (Deng et al., 2024; Jiang et al., 2024). Notably, recent advancements like ARXIV2TABLE (Newman et al., 2024; Wang et al., 2025) have made strides in tabular summarization for scientific literature. However, their primary focus remains on *fuzzy retrieval* and *abstractive summarization* of broad concepts, rather than precise fact extraction. Furthermore, benchmarks like CT-Eval (Shi et al., 2024) and Text-Tuple-Table (Deng et al., 2024) are largely limited to single-document inputs.

5 Conclusion

In this work, we introduced the AOE benchmark to systematically evaluate the end-to-end process of structured knowledge construction from complex, real-world documents. Our comprehensive evaluation of twelve state-of-the-art LLMs reveals that even the best model achieves only 48.8% Cell F1, exposing fundamental capability gaps in numerical reasoning and cross-document synthesis. AOE establishes a challenging new baseline and underscores the need for future research to build truly reliable knowledge extraction systems.

Limitations

While AOE provides a robust framework and valuable insights, several limitations remain that open directions for future work.

Benchmark Scope. AOE, though bilingual (English and Chinese) and spanning four domains, does not cover the full range of languages or complex document types (e.g., medical or engineering records). It currently focuses on flat-table construction, leaving hierarchical table generation for future exploration.

Evaluation. Our evaluation emphasizes factual accuracy via the `Cell F1` metric, but omits other qualitative aspects like schema coherence or conciseness. Efficiency metrics (e.g., latency, cost, API usage) are also excluded, and all results are automatically scored without human verification.

Experimental Scope. The experiments reflect a snapshot of current models under standard prompting settings. Future work could explore agentic frameworks, RAG integration, and broader model configurations to better understand the interplay between retrieval and reasoning in structured extraction tasks.

References

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, and 30 others. 2024. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM.

Anthropic. 2026. The claude model family.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3639–3664.

Amanda Bertsch, Adithya Pratapa, Teruko Mitamura, Graham Neubig, and Matthew R. Gormley. 2025. Oolong: Evaluating long context reasoning and aggregation capabilities. *Preprint*, arXiv:2511.02817.

ByteDance Seed Team. 2026. Seed 2.0 technical report.

Harrison Chase. 2022. LangChain.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Zheye Deng, Chunkit Chan, Weiqi Wang, Yuxi Sun, Wei Fan, Tianshi Zheng, Yauwai Yim, and Yangqiu Song. 2024. Text-tuple-table: Towards information integration in text-to-table generation via global tuple extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9300–9322, Miami, Florida, USA. Association for Computational Linguistics.

Haoyu Dong, Mengkang Hu, Qinyu Xu, Haochen Wang, and Yue Hu. 2024a. Opente: Open-structure table extraction from text. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10306–10310.

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024b. Bamboo: A comprehensive benchmark for evaluating long text modeling capacities of large language models. *Preprint*, arXiv:2309.13345.

Shihan Dou, Yujiong Shen, Chenhao Huang, Junjie Ye, Jiayi Chen, Junzhe Wang, Qianyu He, Shichun Liu, Changze Lv, Jiahang Lin, Jiazheng Zhang, Ming Zhang, Shaofan Liu, Tao Ji, Zhangyue Yin, Cheng Zhang, Huaibing Xie, Jianglu Hu, Jingcheng Deng, and 19 others. 2026a. Cl-bench life: Can language models learn from real-life context? *Preprint*, arXiv:2604.27043.

Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, and 8 others. 2026b. Cl-bench: A benchmark for context learning. *Preprint*, arXiv:2602.03587.

gkamradt. 2025. Needle in a haystack - pressure testing llms.

Google DeepMind. 2026. Gemini 3.1 pro: A smarter model for your most complex tasks.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. 2024. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*.

Peiwen Jiang, Xinbo Lin, Zibo Zhao, Ruhui Ma, Yvonne Jie Chen, and Jinhua Cheng. 2024. TKGT: Redefinition and a new way of text-to-table tasks based on real world demands and knowledge graphs augmented LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16112–16126, Miami, Florida, USA. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

G. Kamradt. 2023. Llmtest: Needle in a haystack. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759.

Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2024a. Loogle: Can long-context language models understand long contexts? *Preprint*, arXiv:2311.04939.

X Li, X Luan, M Zhang, R Wang, J Guo, J Lv, W Qiu, and S Zhao. 2024b. Potential therapeutic option for EGFR-mutant small cell lung cancer transformation: a case report and literature review. *Frontiers in Immunology*, 15:1439033.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2025. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *The Thirteenth International Conference on Learning Representations*.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*.

Moonshot AI. 2026a. Kimi k2.5: Visual agentic intelligence.

Moonshot AI. 2026b. Kimi k2.6: From code to creation, from one to many.

Benjamin Newman, Yoonjoo Lee, Aakanksha Naik, Pao Siangliulue, Raymond Fok, Juho Kim, Daniel S. Weld, Joseph Chee Chang, and Kyle Lo. 2024. Arxivdigestables: Synthesizing scientific literature into tables using language models. *Preprint*, arXiv:2410.22360.

Xuanfan Ni, Piji Li, and Huayang Li. 2023. Unified text structuralization with instruction-tuned language models. *Preprint*, arXiv:2303.14956.

OpenAI. 2024. Hello gpt-4o.

OpenAI. 2026. Introducing gpt-5.4.

Zexuan Qiu, Jingjing Li, Shijue Huang, Xiaoqi Jiao, Wanjun Zhong, and Irwin King. 2024. Clongeval: A chinese benchmark for evaluating long-context large language models. *Preprint*, arXiv:2403.03514.

Qwen Team. 2026. Qwen3.6-plus: Towards real world agents.

Haoxiang Shi, Jiaan Wang, Jiarong Xu, Cen Wang, and Tetsuya Sakai. 2024. Ct-eval: Benchmarking chinese text-to-table performance in large language models. *Preprint*, arXiv:2405.12174.

Mukul Singh, Gust Verbruggen, Vu Le, and Sumit Gulwani. 2024. Tabularis revilio: Converting text to tables. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4056–4060.

Mingyang Song, Mao Zheng, and Xuan Luo. 2024. Counting-stars: A multi-evidence, position-aware, and scalable benchmark for evaluating long-context large language models. *arXiv preprint arXiv:2403.11802*.

Anirudh Sundar, Christopher Richardson, and Larry Heck. 2024. gtbls: Generating tables from text by conditional question answering. *Preprint*, arXiv:2403.14457.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34, Mexico City, Mexico. Association for Computational Linguistics.

Haochen Wang, Kai Hu, Haoyu Dong, and Liangcai Gao. 2024a. Doctabqa: Answering questions from long documents using tables. *Preprint*, arXiv:2408.11490.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024b. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. *Preprint*, arXiv:2406.17419.

Weiqi Wang, Jiefu Ou, Yangqiu Song, Benjamin Van Durme, and Daniel Khashabi. 2025. Can llms generate tabular summaries of science papers? rethinking the evaluation protocol. *Preprint*, arXiv:2504.10284.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. pages 38–45. Association for Computational Linguistics.

Xueqing Wu, Jiacheng Zhang, and Hang Li. 2022. Text-to-table: A new way of information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2518–2533.

Z.ai. 2025. Glm-5.1 - overview - z.ai developer document.

Lei Zhang, Yunshui Li, Ziqiang Liu, Jiaxi Yang, Junhao Liu, Longze Chen, Run Luo, and Min Yang. 2024a. Marathon: A race through the realm of long context with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5201–5217, Bangkok, Thailand. Association for Computational Linguistics.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024b. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Wang Zhao, Dongxiao Gu, Xuejie Yang, Meihuizi Jia, Changyong Liang, Xiaoyu Wang, and Oleg Zolotarev. 2024. Medt2t: An adaptive pointer constrain generating method for a new medical text-to-table task. *Future Gener. Comput. Syst.*, 161:586–600.

.1 Potential Risks for Misuse and Mitigation

In addition to the risks of inaccuracy and misinterpretation discussed in our Ethical Considerations, we acknowledge that the technologies evaluated in this paper could be misused. For example, a highly capable model for structured data extraction could potentially be used to generate misleading or fabricated financial summaries or legal analyses from authentic documents, serving malicious purposes.

However, we believe our primary contribution serves as a form of mitigation against such premature or irresponsible deployment. By systematically demonstrating the significant factual inaccuracies (as shown by the low `Cell F1` scores) and the general unreliability of even state-of-the-art models on this task, our AOE benchmark highlights the current dangers and limitations of these systems. Our findings advocate for caution and underscore the urgent need for further research into factuality and reliability, rather than encouraging the immediate application of these models in high-stakes, real-world domains without rigorous validation.

A Availability and Implementation Details

A.1 License and Availability

The AOE benchmark dataset, along with all source code for our evaluation pipeline, is publicly available at: `https://anonymous.4open.science/r/AOE-Benchmark/`. Both the dataset and the code are released under the **Apache 2.0 License**.

A.2 Implementation and Software Citations

Our experimental framework relies on several key open-source libraries. Deep learning experiments were implemented using **PyTorch** (Ansel et al., 2024). We utilized the **Hugging Face Transformers** (Wolf et al., 2020) library for model loading and inference. Our RAG pipeline was constructed using the **LangChain** (Chase, 2022) framework.

B Compute Resources

All twelve models evaluated in this work are accessed exclusively through their official APIs. Open-weight models (DeepSeek-V3.2, DeepSeek-V4-Pro, Qwen-3.6-Plus, GLM-5.1, Kimi-K2.5, Kimi-K2.6) have publicly available architecture

details and weights documented on their respective model cards and technical reports. Proprietary models (GPT-5.4, Claude-Sonnet-4.6, Claude-Opus-4.6, Gemini-3.1-Pro, Gemini-3-Flash, Seed-2.0-Pro) do not publicly disclose parameter counts.

Inference Hyperparameters. For the Direct and RAG settings, we use `temperature=0` for deterministic decoding. For the Offline Agent and Online Agent settings, we use `temperature=0.1` and `max_tokens=4096` per tool-call turn, with a maximum of 10 agent iterations.

C Generation Settings

We evaluate four generation settings with increasing levels of autonomy. All settings share the same prompt template (query + schema guidance) and output format (CSV); they differ only in how the model accesses source documents.

C.1 Direct (Full-Document Input)

Source documents are concatenated and provided in full as part of the prompt context. The model performs single-pass inference over the entire input. This setting tests the model's core ability to synthesize structured tables from long-context input, with document lengths ranging from 2K to 1.8M tokens across domains.

C.2 RAG (Chunk-Based Retrieval)

Documents are pre-processed into overlapping text chunks (`chunk_size=512`, `overlap=100`) using `RecursiveCharacterTextSplitter`. Each chunk is embedded with BGE-M3 (Chen et al., 2024) and indexed using FAISS (Johnson et al., 2019). At inference time, the top-50 most relevant chunks are retrieved via cosine similarity and provided as context. The pipeline is built with LangChain (Chase, 2022).

- **BGE-M3**: https://huggingface.co/BAAI/bge-m3
- **FAISS**: https://github.com/facebookresearch/faiss
- **LangChain**: https://github.com/langchain-ai/langchain

C.3 Offline Agent (Document Browsing)

An iterative agent that navigates local documents without seeing their full content at once. The agent is equipped with two tools:

- `read_toc(doc_name)`: Returns the table of contents of a document, listing section IDs, headings, and character counts.
- `read_section(doc_name, section_id)`: Returns the full text of a specific section (up to 8,000 characters).

The agent iteratively calls these tools to explore document structure, read relevant sections, and accumulate enough information to produce the target CSV. A maximum of 10 iterations is allowed per task. This setting tests multi-turn reasoning and selective information retrieval from structured documents.

C.4 Online Agent (Web Search)

The model receives *only document names* as input—no document content is provided. The agent must autonomously locate and retrieve information using two external tools:

- `web_search(query)`: Performs a web search via the Serper API[2] and returns the top-10 search result snippets.
- `read_url(url)`: Fetches and extracts the main text content from a URL via Jina Reader[3].

The agent iterates up to 10 rounds of search-and-read cycles to gather sufficient information before generating the final CSV. This setting tests end-to-end information seeking and structuring, simulating scenarios where source documents are not locally available.

- **Serper**: https://serper.dev/
- **Jina Reader**: https://jina.ai/reader/
- **LiteLLM** (unified LLM API): https://github.com/BerriAI/litellm

D Data Collection Details

We curated our dataset exclusively from publicly available, real-world long-form documents. All documents were initially collected in PDF format and then parsed into markdown format to facilitate processing.

[2] https://serper.dev/
[3] https://jina.ai/reader/

Academic Domain We sourced documents from Semantic Scholar[4] and Papers With Code[5]. Semantic Scholar provides a vast repository of academic literature, offering rich metadata on citation contexts and types, which is crucial for tasks requiring analysis of inter-paper relationships. Papers With Code offers leaderboard-style compilations of results for various datasets, providing structured experimental outcomes reported in research papers. All documents collected for this domain were in English.

Financial Domain We collected A-share company annual reports and their summaries of both Chinese and English versions published between 2020 and 2023, sourced from CNINFO[6]. These reports are comprehensive, lengthy, and contain a wealth of structured financial data and textual disclosures, ideal for complex table construction tasks. This domain specifically contributes bilingual (Chinese and English) data to AOE.

Legal Domain We selected Chinese civil law judgments due to their codified nature, unlike precedent-reliant common law systems. This structure is ideal for systematic retrieval, reasoning, and applying statutes. Our dataset draws from two authoritative sources: (1) The People's Court Case Library[7] (providing reference and guiding civil cases) and (2) The China National Database of Laws and Regulations[8] (offering official statutory provisions).

Medical Domain We sourced biomedical case report papers from PubMed Central (PMC) Open Access Subset[9]. The data construction pipeline consists of three stages:

Stage 1: Review Paper Discovery. We queried PubMed with the following search strategy to identify high-quality review articles containing structured case summaries:

```
("case reports"[Publication Type]) AND
("literature review"[Publication Type]
OR "systematic review"[Publication
Type]) OR ("table"[Text Word] OR
"summary of cases"[Text Word]) AND
("open access"[Filter])
```

We restricted results to Q1/Q2 journals (by JCR ranking) to ensure data quality. From the retrieved reviews, we selected those containing "Summary of reported cases" tables—structured tables where each row corresponds to an independent case report paper and columns represent clinical attributes (e.g., age, diagnosis, treatment, outcome).

Stage 2: Source Paper Collection and Filtering. For each row in a selected review table, we retrieved the original case report paper cited therein. Two filtering criteria were applied: (1) the cited paper must be Open Access (ensuring legal redistribution), and (2) each markdown document must describe exactly one clinical case (ensuring a clean one-row-per-document mapping). Papers failing either criterion were excluded, and the corresponding rows were removed from the ground-truth table.

Stage 3: Human Annotation and Quality Control. Annotators verified each instance to ensure: (a) no duplicate rows exist, (b) column names are clinically meaningful and consistent across instances within the same review, (c) the number of target rows is manageable (3–15 per instance), and (d) cell values in the ground-truth table are traceable to specific content in the source papers. The review table itself serves as the gold standard, leveraging the domain expertise of the original review authors.

This pipeline yields ground-truth tables that are inherently authoritative—they are curated by domain experts (the review paper authors) and published in peer-reviewed journals, providing a stronger quality guarantee than crowd-sourced annotations.

E Detailed Annotation Process with Examples

To further illustrate the two-phase annotation process described in the main text, this section provides a concrete, step-by-step example from the Academic domain. This walkthrough demonstrates how we ensure the accuracy and consistency of the ground-truth data in the AOE benchmark.

E.1 Annotator Information

Our annotation process involved a multi-stage, expert-driven approach. The core annotation team consisted of graduate students from our institution with relevant domain expertise (e.g., in finance and law). In recognition of their significant intellectual contributions to the benchmark's creation—which included refining task definitions, developing annotation guidelines, and performing the initial data

[4] https://www.semanticscholar.org/
[5] https://paperswithcode.com/
[6] https://static.cninfo.com.cn/
[7] https://rmfyalk.court.gov.cn
[8] https://flk.npc.gov.cn/index.html
[9] https://pmc.ncbi.nlm.nih.gov/

annotation and validation—they are credited as co-authors on this paper.

Additionally, to ensure the highest level of accuracy for specialized content, we recruited external domain experts (practitioners in the financial and legal fields) to conduct a final round of review and adjudication on complex cases. These experts were compensated for their time at a rate commensurate with their professional expertise.

E.2 Example: Annotating Methodology Performance in the Academic Domain

- **Context:** The source documents are several computer vision research papers that report performance metrics on the MS COCO dataset.

- **Pharse 1: Expert-led Query Definition**

  A domain expert with a background in computer science formulates a precise, real-world query to guide the extraction process.

  > **Defined Query:** "List the FID performance of the proposed methods in the paper on the MS COCO dataset."

  The expert then designs a structured table schema to hold the extracted information, defining the necessary columns and their expected content. This schema is then provided to the annotators.

  > **table_schema**
  > {"columns": [
  > {"name": "paper_name", "about": "Paper Title"},
  > {"name": "method", "about": "Proposed Method"},
  > {"name": "result", "about": "Performance on Specific Metrics"},
  > {"name": "models_and_settings", "about": "Model Settings"}
  >]}

- **Pharse 2: Dual Annotation and Final Adjudication**

  The source documents and the empty table schema are assigned to two separate annotators. Each annotator independently reads the papers to find the relevant values and populates the Table 6. If the results were identical, the annotation was accepted. If there were discrepancies, the task was escalated to a senior reviewer for final judgment.

paper_name	method	result	settings
Paper A	GAN	21.5	Standard settings (Sec. 3)
Paper B	StyleGAN	4.40	Style mixing, 1024^2, no trunc.
Paper C	ADM	2.97	Classifier guide, 250 DDPM steps

Table 6: Final adjudicated ground-truth table after the annotation process.

This structured, multi-stage process is applied across all domains in our benchmark to ensure the highest possible data quality.

F Details of Tasks definition

Unlike conventional benchmarks where each document maps to a single output row, our benchmark employs a **dynamic row generation** strategy where the relationship between input documents (M) and output rows (R_T) is task-dependent and reflects real-world analytical requirements.

F.1 Design Principles

Inferential Reasoning Requirements Tasks like $Legal_0$ (Figure 5, up) demand semantic understanding beyond keyword matching. Models must interpret legal narratives and map them to precise statutory provisions through causal reasoning—a capability that distinguishes genuine comprehension from pattern recognition.

Atomic Granularity and Practical Value By requiring extraction of structured, atomic data points rather than aggregated summaries, tasks like $Legal_1$ support high-value downstream applications (e.g., quantitative disparity analyses) while preventing degradation into conventional summarization.

Table 16 presents the complete query schemas and ground truth columns for all tasks, demonstrating how these principles manifest across domains.

F.2 Row Generation Patterns Across Domains

One-to-Many Mapping (Legal Domain). In tasks such as $Legal_1$**: Defendant Verdict Extraction** (Figure 5, bottom), a single case document may yield multiple output rows when the case involves multiple defendants. Each row atomically

Input	Example Content
Case Facts	On July 4, 2013, Zou Moumou, Qi Mou, and Yu Moumou stole materials to manufacture drugs...
Legal Docs	*Full text of "Criminal Law of the PRC"*
Output	`["Criminal Law Article 23", "Article 347"]`

Design Principle: Inferential Reasoning

Requires **deep legal reasoning** to map narrative facts to precise statutory provisions—not achievable through keyword matching.

Case	Defendant	Charge	Term
Guan Case	Guan M.	Embezzlement	8 yrs
Guan Case	Zhao M.	Embezzlement	1.5 yrs
Xu Case	Xu M.	Embezzlement	3.75 yrs

Design Principle: Structured Atomicity

Extracts **granular, structured data** enabling quantitative analysis (e.g., sentencing studies) rather than summarization.

Figure 5: Legal domain task design principles. **Top:** $Legal_0$ demonstrates *inferential reasoning*—models must understand legal narratives and map them to precise statutory provisions through semantic understanding. **Bottom:** $Legal_1$ exemplifies *atomic granularity*, where a single case document produces multiple rows when multiple defendants are involved, enabling quantitative analyses.

captures verdict details (defendant name, charge, sentence term) for individual defendants, enabling downstream quantitative analyses such as sentencing disparity studies across demographic or jurisdictional factors.

Document-to-Citation Mapping (Academic Domain). For Aca_0**: Citation Context Extraction**, rows represent unique citation linkages between documents. A single paper citing k references generates k rows, each containing the citation context and metadata. This design supports bibliometric analyses and citation pattern studies.

Temporal Decomposition (Financial Domain). Longitudinal financial tasks (Fin_0–Fin_3) adopt a temporal row structure where each row corresponds to a specific reporting period. For instance, analyzing a company's 5-year performance from a single document containing multi-year data produces $R_T = 5$ rows, with each row capturing period-specific metrics.

Cross-Entity Aggregation (Financial Domain). In **cross-company benchmarking tasks** (Fin_4–Fin_6), each row represents a distinct entity. Analyzing M companies' annual reports yields $R_T = M$ rows, enabling comparative ratio analyses and peer benchmarking.

G RAG Setting: Implementation and Length-Based Analysis

G.1 Implementation Details

Our RAG pipeline follows a standard retrieve-then-generate paradigm. Source documents are first split into overlapping text chunks (`chunk_size=512`, `overlap=100`) using `RecursiveCharacterTextSplitter`. Each chunk is embedded with BGE-M3 and indexed using FAISS. At inference time, the query (task description + schema guidance) is used to retrieve the top-50 most relevant chunks via cosine similarity, which are then concatenated and provided as context for the language model. The entire pipeline is built with LangChain.

G.2 Performance by Document Length

To systematically investigate why RAG degrades on complex extraction tasks, we construct a standardized evaluation subset of 140 cases, evenly split into **short** (70 cases) and **long** (70 cases) document groups. Table 7 reports the full results.

Across all evaluated models, RAG performance drops substantially on the long subset. Cell F1 decreases by 10–17 absolute percentage points: Claude-Opus-4.6 falls from 59.5% (short) to 41.8% (long), while DeepSeek-V3.2 drops from 54.9% to 37.8%. The fine-grained one-piece scores (Scores 1–4) exhibit a consistent downward shift, with GPT-5.4's mid-level scores (Score 2 and Score 3) nearly halving from 34.1% and 31.9% to 18.7% and 16.5%, respectively.

The root cause is twofold: (1) as document length grows, the retrieval index expands with more chunks, lowering the signal-to-noise ratio in the retrieved context; (2) multi-hop dependencies—where the evidence for a single cell spans multiple distant passages—are structurally severed by chunking, making it impossible for the retriever to recover the full reasoning chain in a single top-k pass.

Table 7: Performance of the Retrieval-Augmented Generation (RAG) setting across different source document lengths. The evaluation set consists of 140 cases, evenly divided into *short* (70 cases) and *long* (70 cases) subsets. The noticeable performance drop in the long subset highlights the retrieval and noise challenges inherent to RAG systems when processing extensive documents. All values in %.

Model	Split	Primary Metrics			One-Piece Scores			
		Cell F1	Col F1	Row F1	Score 1	Score 2	Score 3	Score 4
CLAUDE-OPUS-4.6	Short	59.5	62.0	98.5	74.9	67.6	62.3	81.3
	Long	41.8	51.7	92.9	57.2	48.6	41.6	63.6
DEEPSEEK-V3.2	Short	54.9	57.4	98.5	52.3	43.5	40.9	66.6
	Long	37.8	48.9	88.0	27.7	16.5	14.1	43.1
DEEPSEEK-V4-PRO	Short	44.7	53.8	80.4	73.5	65.2	60.8	87.0
	Long	27.9	42.5	69.1	64.7	57.0	47.9	82.5
GEMINI-3.1-PRO	Short	55.0	53.2	100.0	74.3	65.6	65.3	87.0
	Long	43.8	46.6	100.0	66.3	56.7	50.8	77.6
GPT-5.4	Short	53.4	62.8	86.2	44.3	34.1	31.9	49.7
	Long	36.1	56.1	76.4	28.8	18.7	16.5	43.0
GPT-5.4-0305	Short	58.5	62.9	94.6	75.1	67.4	63.0	82.2
	Long	41.6	52.6	89.3	65.2	56.9	48.5	73.0
QWEN-3.6-PLUS	Short	54.6	56.3	100.0	74.2	66.2	63.5	84.2
	Long	38.7	46.7	90.5	56.0	47.0	41.1	58.7

G.3 Case Study: Semantic vs. Factual Retrieval

Table 8 illustrates a representative failure. When queried for "shareholder value creation analysis," RAG retrieves semantically related but factually useless passages (e.g., board statements about "commitment to shareholder value"), while missing the numerical evidence (earnings per share, net assets) that is required to populate the target table. This highlights a fundamental mismatch: standard embedding-based retrieval optimizes for topical similarity, whereas structured extraction demands precise factual recall—a gap that widens with document length.

Query: Assess "shareholder value creation analysis"

Required Evidence	Context Retrieved by RAG
`Earnings per share (basic): $1.15; Net assets per share: $12.50; Total equity...`	`"Our board of directors remains fully committed to creating long-term shareholder value through strategic investments..."`

Table 8: A RAG failure case. RAG retrieves semantically related passages (topical relevance) but misses the numerical evidence (factual precision) required for structured extraction.

H Prompt Examples

We use the following prompt format. The full prompt specifications and usage examples are provided in our supplementary materials/code repository.

H.1 Prompt Examples for Generating Tables

H.1.1 Prompt Example of Offline Setting for LLMs

```
Instruction:
    Role: Academic Citation Relationship
     Analyzer
```

```
    Objective: Extract and analyze citation
     relationships between academic papers from
     Markdown files.

    # Workflow Steps
    1. Document_Parsing:
       - Extract paper title from H1 header (e.g
    ., "# Title")
       - Identify citation markers: "(Author et
     al., YYYY)" or "[N]"

    2. Table_Schema_Design:
       - Columns: ["Referencing_Paper_Title", "
    Cited_Paper_Title",
                   "Citation_Context", "
    Citation_Purpose"]
```

```
    3. Citation_Linking:
       - Match citation markers to reference
    entries
       - Extract 2 sentences before/after
    citation
       - Classify purpose: ["background", "
    methodology",
         "comparison", "data_source", "
    supporting_evidence"]

    4. Data_Structuring:
       - Output flat CSV table
       - Wrap fields with commas/quotes in
    double quotes

    Query: {query}
    Input_doc: {input_doc}

    Output Format:
    <schema>
    Schema:
      - name: Cited_Paper_Title
        valid_type: exact_match
      - name: Referencing_Paper_Title
        valid_type: exact_match
      - name: Citation_Context
        valid_type: fuzzy_match
      - name: Citation_Purpose
        valid_type: categorical
    </schema>

    ‘‘‘csv
    Cited paper title,Referencing paper title,
    Citation context,Purpose
    "Paper A","Paper B","...context1...",
    background
    "Paper A","Paper B","...context2...",
    background
    "Paper B","Paper A","...context3...",
    background
    ‘‘‘
```

H.1.2 Prompt Example of Online Settings for Agents and DeepReasearch

```
Instruction:
    You are a financial data analysis expert.
    Based on the given Query and Input_doc,
    extract specified information from financial
     statements and output it in CSV format.
    Please follow the steps below and output the
     results.

    **Step 1: Understand the task requirements
    and analyze the input data.**
    Since the Input_doc contains only a list of
    filenames, treat this as an online scenario.
     You must use the web search function to
    find and locate the content of these
    documents. Briefly describe your search
    strategy and the key sources found in the <
    search_summary> section.

    **Step 2: Design the output table structure
    .**
    Design a Schema based on the Query’s
    requirements, defining the column names and
    data types to ensure a standardized output
    format.

    **Step 3: Implementation.**
    Based on the analyzed document content from
    your web search results, extract the content
     relevant to the Schema headers, strictly
    adhering to the format and unit requirements
     of the Schema, and output a single-level
    table in CSV format.

    **Step 4: Format check.**
    Enclose fields containing special characters
     like commas or newlines in double quotes.
    In specific cases, consider replacing commas
     with semicolons, spaces, or removing them
    entirely, provided it does not alter the
    original meaning of the data.

    **When processing financial data, please
    note:**
    - Convert monetary amounts to the standard
    unit (e.g., 4.5 million -> 4,500,000).
    - Keep the unit for share capital as "shares
    ".
    - Extract the numerical value from
    percentages (e.g., 12.36% -> 12.36).
    - Convert negative values represented by
    parentheses to a negative number (e.g.,
    "(1,234)" -> -1234).
    - Retain thousand separators in numerical
    values and enclose the field in double
    quotes (e.g., "1,234,567").
    - Round monetary values to the nearest whole
     number and ratios to two decimal places.
  Instruction_no_cot: |
    You are a financial data analysis expert.
    Based on the given Query and Input_doc,
    extract specified information from financial
     statements and output it in CSV format. If
    the Input_doc only contains filenames, use
    web search to find the document content.

  Output: |
    Query: {query}
    Input_doc: {input_doc}
    <search_summary>
     Briefly summarize your search strategy and
    the key document sources found here.
    </search_summary>
    Answers:
      <schema>
       The schema you built
      </schema>
      ‘‘‘csv
       Your Output
      ‘‘‘

  Example: |
    Example 1:
      Query: "For the specified documents,
    conduct a corporate profitability analysis."
      Input_doc:
        - "BBB_Group_2022_Q3_Report.pdf"
        - "BBB_Group_2022_Annual_Report.pdf"
      Answers:
        <search_summary>
         Performed a web search for "BBB Group
    2022 Q3 Report" and "BBB Group 2022 Annual
    Report". Found the PDF versions of both
    reports on the official stock exchange’s
    information disclosure website. Data was
```

```
    primarily extracted from the consolidated
    income statements.
        </search_summary>
        <schema>
          - name: Document Name
            about: Full filename of the
    financial report, string type.
          - name: Reporting Period
            about: The accounting period covered
     by the report (YYYY-Qn format).
            format: 2021-Q4
          - name: Total Operating Revenue ($)
            about: The value of the first line
    item in the income statement.
            valid_type: exact_match
          - name: Gross Margin (%)
            about: (Operating Revenue -
    Operating Cost) / Operating Revenue * 100,
    rounded to two decimal places.
          - name: Net Profit Attributable to
    Parent ($)
            about: Net profit attributable to
    the parent company from the consolidated
    income statement.
            valid_type: exact_match
          - name: Net Margin (%)
            about: Net Profit Attributable to
    Parent / Total Operating Revenue * 100,
    rounded to two decimal places.
          - name: Non-recurring P/L Ratio (%)
            about: Non-recurring profit or loss
    / absolute value of Net Profit * 100,
    rounded to two decimal places.
        </schema>
        ‘‘‘csv
        Document Name,Reporting Period,Total
    Operating Revenue ($),Gross Margin (%),Net
    Profit Attributable to Parent ($),Net Margin
     (%),Non-recurring P/L Ratio (%)
        "BBB_Group_2022_Q3_Report.pdf",2022-Q3
    ,"88,456,789,234",32.15,"9,345,678,123",
        10.56,2.34
        "BBB_Group_2022_Annual_Report.pdf",2022-
    Q4,"345,678,901,567",30.89,"38,901,234,567",
        11.25,1.07
        ‘‘‘
```

H.2 Prompt Examples for Evaluating Tables

H.2.1 Prompt Examples for LLM Overall Scores

```
Instruction: |
  # Evaluation Task Description
  As a professional evaluator, strictly compare
     the CSV output quality of Gold Answer and
     Predict Answer based on the Query.
     Evaluation dimensions include:

  ## Core Evaluation Dimensions

  1. Intent Understanding: <score1>{int}</score1
     >/100
       Evaluate the accuracy of Predict Answer’s
     understanding of the data extraction or
     organization requirements in the Query. The
     core is whether the model grasps the Query’s
      intent, identifies the types of data points
      to be extracted or organized, their
     relationships, and the core information the
     final table should reflect.
       * **Perfect Score (100):** Complete and
     accurate understanding of all Query
     requirements, with no deviations or
     omissions. Can clearly identify core tasks.
       * **High Score (80-99):** Accurate
     understanding of the main Query intent, but
     may have slight misunderstandings on minor
     details or implicit requirements that don’t
     affect core task execution.
       * **Medium-Low Score (40-79):** Partial
     understanding bias or confusion about the
     core Query intent, capturing some
     information points but failing to fully
     distinguish different types of information.
       * **Low Score (1-39):** Serious
     understanding bias or fundamental
     misunderstanding of the core Query intent,
     directing the task in a completely different
      direction.
       * **Zero Score (0):** Output completely
     unrelated to Query requirements, or unable
     to determine any understanding of the Query.
       Score Range: 0-100

  2. Schema Construction: <score2>{int}</score2
     >/100
       Evaluate the completeness and accuracy of
     the Schema (column names/headers)
     constructed by Predict Answer. Whether the
     model correctly identifies and includes all
     necessary information fields required by the
      Query, excludes irrelevant fields, and
     whether field naming is clear and accurate.
       * **Perfect Score (100):** Constructed
     Schema completely matches Query requirements
      or Gold Answer structure, includes all
     necessary columns, with no redundant or
     incorrect columns.
       * **High Score (80-99):** Schema basically
      complete and correct, includes most
     necessary columns, may miss 1-2 minor
     columns.
       * **Medium-Low Score (40-79):** Schema has
      significant deficiencies, missing multiple
     important columns, or contains many
     irrelevant/incorrect columns.
       * **Low Score (1-39):** Schema extremely
     incomplete or error-ridden, missing almost
     all key columns.
       * **Zero Score (0):** Failed to generate
     any Schema (no headers), or generated Schema
      completely unrelated to Query requirements.
       Score Range: 0-100

  3. Content Accuracy: <score3>{int}</score3
     >/100
       Evaluate the correctness of **specific
     data content** filled in the table by
     Predict Answer. Based on the Schema
     constructed by Predict Answer, check whether
      the data values of each filled field are
     precisely consistent with the information in
      Input Doc or Gold Answer.
       * **Perfect Score (100):** All filled data
      values in the table accurately and error-
     free reflect the corresponding information
     in Input Doc.
```

```
      * **High Score (80-99):** Most data values
     are correct, with only minor errors (such
    as spelling errors, extra spaces, etc.).
      * **Medium-Low Score (40-79):** Many data
    values are inaccurate, or extracted data is
    related but not precise enough.
      * **Low Score (1-39):** Most filled data
    values are incorrect, irrelevant, or
    completely fabricated (hallucinations).
      * **Zero Score (0):** All data filled in
    the table is incorrect, irrelevant, or
    completely fabricated.
      Score Range: 0-100

  4. Format Compliance: <score4>{int}</score4
    >/100
      Evaluate whether the output file of
    Predict Answer strictly follows the format
    requirements specified in the Query or task
    description, especially CSV basic format (
    delimiters, quote usage, line breaks, file
    encoding, etc.).
      * **Perfect Score (100):** Output file
    strictly adheres to all format
    specifications, file can be read without
    errors by standard parsers.
      * **High Score (80-99):** Output file
    basically meets format requirements, can be
    read by standard parsers, but has minor
    format issues.
      * **Medium-Low Score (40-79):** Output
    file has format issues that cause partial
    parsing errors or warnings.
      * **Low Score (1-39):** Overall format of
    output file does not meet requirements (such
     as outputting Markdown, plain text, etc.
    instead of CSV format).
      * **Zero Score (0):** No output file
    generated, or output is completely
    incomprehensible gibberish.
      Score Range: 0-100

  ## Score Parsing Markers
  Final output must include parsable delimiters,
      and scores must be integers:
  - <score1>{int}</score1>  # Corresponds to
    Intent Understanding
  - <score2>{int}</score2>  # Corresponds to
    Schema Construction
  - <score3>{int}</score3>  # Corresponds to
    Content Accuracy
  - <score4>{int}</score4>  # Corresponds to
    Format Compliance

Output: |
    Query: {query}

    Gold Answer csv: {gold_csv_txt}
    Predict Answer csv: {predict_csv_txt}

    Output:
    <output>
      1. Intent Understanding: <score1>score1</
    score1>/100 | Reason:
      2. Schema Construction: <score2>score2</
    score2>/100 | Reason:
      3. Content Accuracy: <score3>score3</
    score3>/100 | Reason:
      4. Format Compliance: <score4>score4</
    score4>/100 | Reason:
    </output>

Examples: |
    Example 1 - Academic Citation Extraction:
      Query: "Identify possible citation
    relationships among the following articles,
    understand, and provide the specific content
     being cited from the cited article."

      Gold Answer csv:
        Cited paper title,Referencing paper
    title,Referenced content,Label
        More Than Spoken Words: Nonverbal
    Message Extraction and Generation,ConceptNet
     5.5: An Open Multilingual Graph of General
    Knowledge,"Thus, we use the human-annotated
    argument pairs such as (someone holds an
    umbrella, it is raining) in the Chinese set
    of commonsense knowledge ConceptNet by
    regarding the two arguments as the input and
     the NM respectively.",methodology
        More Than Spoken Words: Nonverbal
    Message Extraction and Generation,Know What
    You Don't Know: Unanswerable Questions for
    SQuAD,"an extractor similar to the
    formulation and implementation to address
    the unanswerable questions in machine
    reading comprehension tasks or generator
    simply using a pre-defined answer such as
    empty or none as the NM of these instances",
    background

      Predict Answer csv:
        Referencing Paper Title,Cited Paper
    Title,Citation Context,Citation Purpose,
    Citation Marker
        More Than Spoken Words: Nonverbal
    Message Extraction And Generation,Breazeal
    et al. (2005),"Nonverbal messages (NM), such
     as facial expressions, body movements, and
    tones of voice, can complement or modify
    verbal messages as well as improve the
    teamwork efficiency and effectiveness of
    face-to-face communication.",background,(
    Breazeal et al. 2005)
        More Than Spoken Words: Nonverbal
    Message Extraction And Generation,Phutela
    (2015),"Nonverbal messages (NM), such as
    facial expressions, body movements, and
    tones of voice, can complement or modify
    verbal messages as well as improve the
    teamwork efficiency and effectiveness of
    face-to-face communication.",background,(
    Phutela 2015)
        More Than Spoken Words: Nonverbal
    Message Extraction And Generation,Csaky and
    Recski (2021),"NMs are also seldom included
    in existing text-based dialogue tasks that
    mainly focus on verbal messages.",background
    ,(Csaky and Recski 2021)

      Output:
      <output>
        1. Intent Understanding: <score1>40</
    score1>/100 | Reason: Query requires "
    Identify possible citation relationships" (
    identify citation relationships), "
    understand" (understand), and "provide the
    specific content being cited from the cited
    article" (provide specific content from
```

```
    cited article).
        2. Schema Construction: <score2>30</
    score2>/100 | Reason: The Schema constructed
     by the model (Referencing Paper Title,
    Cited Paper Title, Citation Context,
    Citation Purpose, Citation Marker) deviates
    from the core Query requirement (extract "
    referenced content").
        3. Content Accuracy: <score3>20</score3
    >/100 | Reason: Query requires providing
    accurate "specific content from the cited
    article". The "Citation Context" provided in
     the prediction is text from the citing
    paper, not precise content from the cited
    article, which is completely inaccurate in
    content type.
        4. Format Compliance: <score4>95</score4
    >/100 | Reason: Output is standard CSV
    format, using commas as delimiters, data
    rows correspond to headers. Format
    compliance is very high.
      </output>
```

H.2.2 Prompt of Column Mapping and Format checking from LLMs

```
mapping_predict_tables:
    Instruction: |
      step1:
        Analyze column name matching
        Find the correspondence between the
    Gold_column_names and the Predict_answers
    column names through analysis and semantic
    matching. This includes case-insensitive
    matching and matching aliases or synonyms.

        For example:
        'Reporting Period' is equivalent to '
    Fiscal Year'.
        'Owner's Equity' is equivalent to 'Net
    Assets'.
        'Referenced Content' is equivalent to '
    Citation Context'.
        'Corresponding Legal Provisions' is
    equivalent to 'Related Index'.
        'Label' is equivalent to 'Citation
    Purpose'.

        Hint: If a column name in
    Predict_answers is too general, it can be
    split based on the more specific column
    names found in Gold_column_names.

        The finally determined matching columns
    **intersection_list**, ensure that each item
     name in it comes from Gold_column_names.
      step2:
        Ensure that each item in the
    intersection_list comes from
    Gold_column_names.
      step3:
        Use the intersection_list as the CSV
    header names, and fill the corresponding
    content from Predict_answers into this CSV.
      step4:
        Format check, ensure the generated CSV
    can be processed by pd.read_csv. If numbers
    contain commas as thousand separators, this
    will cause CSV parsing issues. To ensure
    correct processing by `pd.read_csv`, these
    numbers with commas need to be enclosed in
    quotes.

    Output: |
      Input:
        Predict_answers:
        ```csv
         {predict_txt}
        ```
        Gold_column_names:
         {gold_header_string}

      Output:
        I have already mapped predict_txt
    according to the gold_header_string column
    names. Below is the processed Predict_csv:
        ```csv
        [Complete corrected CSV]
        ```
```

H.2.3 Prompt Examples for Mapping Evaluation (Cell F1)

```
Instruction: |
  You are an expert evaluator strictly comparing
     text extracted into a table cell based ONLY
     on the provided Gold Standard and Predicted
     Text for the given Column Name. Adhere ONLY
     to the rules below.

  Evaluation Rules:
    1. Exact/Semantic Match: If the Predicted
    Text perfectly matches the Gold Standard
    Text OR conveys exactly the same essential
    information (allowing for minor formatting
    differences like currency symbols/names,
    date formats 'YYYY-MM-DD' vs 'Month D, YYYY
    ', synonyms for common words that do not
    change meaning, or ignoring harmless extra
    punctuation like a trailing comma), output a
     score of 1.0.

    2. Critical Error/Missing Key Info: If the
    Predicted Text contains factually incorrect
    information compared to the Gold Standard,
    is completely unrelated, OR misses crucial
    information present in the Gold Standard,
    output a score of 0.0.

    3. List/Array Content (if applicable): If
    the content represents a list of items (
    detectable by structure like ["a","b"], a, b
    , c, or line breaks):
        - Identify the distinct items in the
    Gold Standard list (count = N_gold).
        - Identify the distinct items in the
    Predicted list (count = N_pred).
        - Find the number of items present in
    both lists (count = N_correct). Treat items
    case-insensitively and ignore minor
    whitespace differences for matching list
    items.
        - Calculate Score = (N_correct / N_gold)
     * 0.8. If N_gold is 0, the score is 1.0 if
    N_pred is also 0, otherwise 0.0. Ensure
    score does not exceed 0.8.
        - Note: If Gold is ["A", "B"] and
    Predicted is ["A", "C"], N_correct=1, N_gold
```

```
    =2, score = (1/2)*0.8 = 0.4.
        - If the field is NOT list-like, apply
    rules 1, 2, or 4.

    4. Partial Match (Other Cases): If it's not
    a perfect match (Rule 1), not a critical
    error (Rule 2), and not a list (Rule 3),
    estimate the degree of semantic overlap and
    correctness. For example, if one minor
    detail is wrong but the rest is correct,
    assign a score between 0.5 and 0.9. If
    significant parts are correct but some key
    info is missing or wrong, assign 0.1 to 0.5.
     Use your judgment based on how much of the
    core information is preserved.

    5. Empty/Null Cases:
        - If Gold is empty/null/NA and Predicted
     is also empty/null/NA, score is 1.0.
        - If Gold is empty/null/NA but Predicted
     has content, score is 0.0.
        - If Gold has content but Predicted is
    empty/null/NA, score is 0.0.

Output: |
  Column Name: {column_name}
  Gold Standard: "{gold_text}"
  Predicted Text: "{predicted_text}"
  Match Score: <output>0|1|0.x</output>
  Note: No explanation needed, output directly
     as <output>0|1|0.x</output>

Examples: |
  Examples (Illustrative based on rules):
    # Rule 1 Example
    - Column: 'Product Price'
      Gold: '"$19.99"'
      Predicted: '"19.99 USD"'
      Score: <output>1.0</output>

    # Rule 2 Example
    - Column: 'Delivery Date'
      Gold: '"2023-05-15"'
      Predicted: '"May 16, 2023"'
      Score: <output>0.0</output>

    # Rule 3 Example (List)
    - Column: 'Legal Articles'
      Gold: '["Article 14","Article 39","Article
     40"]' # N_gold = 3
      Predicted: 'Article 14, Article 45'
            # N_pred = 2, N_correct = 1 ("Article
     14")
      Score: <output>0.27</output>
            # (1/3) * 0.8 -> 0.27

    # Rule 3 Example (List)
    - Column: 'Related References'
      Gold: '["Patent CN2019/001","DOI 10.1234"]'
     # N_gold = 2
      Predicted: 'Patent CN2019/001'
       # N_pred = 1, N_correct = 1
      Score: <output>0.4</output>
       # (1/2) * 0.8 = 0.4

    # Rule 4 Example (Tolerance)
    - Column: 'Industry Ban Period'
      Gold: 'Prohibited from transportation
    industry for 4 years'
      Predicted: ', Prohibited from
    transportation industry for four years'
      Score: <output>1.0</output>

    - Column: 'Probation'
      Gold: 'Probation for one year and six
    months'
      Predicted: 'One year and six months'
      Score: <output>1.0</output>

    - Column: 'Other Rulings'
      Gold: 'None'
      Predicted: ''
      Score: <output>1.0</output>

    # Rule 4 Example (Partial)
    - Column: Description
      Gold: "Red widget model X-1, uses 2 AA
    batteries."
      Predicted: "Red widget version X-1,
    requires batteries." # Missing battery type/
    count
      Score: <output>0.7</output> # Subjective
    partial score

    # Rule 5 Example both empty
    - Column: Middle Name
      Gold: ""
      Predicted: ""
      Score: <output>1.0</output>
```

I Algorithm of Content Evaluation

The algorithm of content evaluation can be seen at algorithm 1.

J Stability and Variance Analysis

To evaluate the robustness of our benchmark, we conduct a multi-run analysis: five representative LLMs (Claude-Opus-4.6, DeepSeek-V3.2, GPT-5.4, Kimi-K2.5, Qwen-3.6-Plus) are each evaluated across 5 independent runs. All metrics use macro-averaging (per-instance scores averaged over all instances; CSV-failed instances receive Cell F1 = 0).

J.1 Run-to-Run Score Stability

Table 9 reports per-model, per-domain statistics across all four key metrics. The majority of model-domain combinations exhibit Cell F1 standard deviations below 3 percentage points, confirming that performance differences are statistically meaningful. Kimi-K2.5's Academic domain shows the highest variance (SD = 7.6%), attributable to occasional CSV formatting failures that produce run-level score swings.

J.2 Ranking Stability

Table 10 examines whether model *rankings* are stable across runs—a stricter test than absolute score stability.

Model	Domain	Pass Rate	Col Recall	Row F1	Cell F1
Claude-Opus-4.6	Financial	90.5 ± 6.0	69.3 ± 1.7	83.6 ± 4.0	45.9 ± 1.2
	Legal	93.2 ± 4.1	96.8 ± 1.1	72.4 ± 1.4	37.7 ± 1.8
	Academic	100.0 ± 0.0	100.0 ± 0.0	57.0 ± 0.8	39.1 ± 1.7
DeepSeek-V3.2	Financial	90.0 ± 3.2	64.6 ± 1.8	85.2 ± 2.0	42.4 ± 1.0
	Legal	93.8 ± 2.8	91.8 ± 2.2	73.4 ± 2.3	43.6 ± 2.3
	Academic	80.2 ± 3.3	100.0 ± 0.0	45.9 ± 1.9	37.8 ± 1.2
GPT-5.4	Financial	93.9 ± 3.3	68.9 ± 2.4	87.3 ± 3.8	44.6 ± 3.1
	Legal	93.1 ± 4.2	97.4 ± 0.6	65.5 ± 3.4	38.3 ± 2.7
	Academic	100.0 ± 0.0	99.6 ± 0.4	64.2 ± 1.1	47.7 ± 1.3
Kimi-K2.5	Financial	93.8 ± 4.0	66.3 ± 2.0	80.1 ± 3.9	41.5 ± 4.4
	Legal	92.3 ± 3.2	95.4 ± 0.8	75.5 ± 6.3	43.1 ± 4.8
	Academic	97.2 ± 1.4	99.4 ± 0.7	62.9 ± 8.5	47.6 ± 7.6
Qwen-3.6-Plus	Financial	93.8 ± 1.0	59.6 ± 2.6	83.1 ± 3.1	37.7 ± 2.3
	Legal	83.8 ± 4.7	91.4 ± 3.1	68.2 ± 5.8	41.7 ± 3.8
	Academic	96.6 ± 1.3	99.7 ± 0.6	66.9 ± 1.2	50.7 ± 1.1

Table 9: Multi-run performance (Mean ± SD) across 5 independent runs per model. CSV-failed instances score 0 and are included in the denominator. All values in %.

Model	R0	R1	R2	R3	R4	Range
Qwen-3.6-Plus	1	3	3	2	2	1–3
Kimi-K2.5	3	1	1	4	1	1–4
GPT-5.4	2	2	2	1	3	1–3
DeepSeek-V3.2	4	5	4	3	5	3–5
Claude-Opus-4.6	5	4	5	5	4	4–5

Table 10: Overall rankings (macro-averaged Cell F1 across 3 domains) per run. Kendall's $W = 0.664$. The top tier (Qwen, Kimi, GPT) and bottom tier (DeepSeek, Claude) are distinguishable, but within-tier ordering fluctuates.

Value Type	Mean Cell F1	SD
Numeric	49.9	1.9
Calculation	41.8	2.1
Reasoning	29.6	3.4
Text	28.8	1.6

Table 11: Cell F1 (%) by answer value type, averaged across 5 models × 5 runs. Low SD confirms metric stability across value categories.

The overall concordance ($W = 0.664$) reflects moderate ranking stability. The top three models (Qwen-3.6-Plus, Kimi-K2.5, GPT-5.4) consistently outperform the bottom two (DeepSeek-V3.2, Claude-Opus-4.6), but within-tier ordering varies across runs due to tight score clustering (top-tier mean: 43–44%, bottom-tier: 41%). Per-domain concordance ranges from $W = 0.464$ (Financial) to $W = 0.712$ (Legal), indicating that Legal rankings are the most robust while Financial rankings are least stable.

J.3 Value Type Stability

Table 11 reports Cell F1 by answer value type, averaged across all 5 models and 5 runs.

Numeric cells achieve the highest Cell F1 (49.9%), as they involve constrained value lookup. Calculation cells score lower (41.8%), reflecting multi-step arithmetic difficulty. Reasoning and text cells prove most challenging (∼29%), requiring cross-document inference or open-ended synthesis respectively.

K Schema Guidance: Case Study

Table 12 illustrates the dominant failure mode. Without schema guidance, Claude-Sonnet-4.6 produces a table that is *clinically sound*—it extracts age, sex, disease subtype, and diagnostic sequence, all factually correct. However, this "ideal" schema shares only 2 columns with the gold standard (Treatment, Outcome), making 60% of extracted information invisible to evaluation. Worse, the model disaggregates the gold's summary-level rows (e.g., "2 cases of TAK+CD") into individual patient entries, causing row-level misalignment.

This case demonstrates why schema guidance is not merely an evaluation convenience but a *design necessity*: without it, models and ground-truth operate in different semantic spaces, making Cell F1 an unreliable signal of extraction quality.

L Human Performance

To contextualize model performance, we conducted a human annotation experiment on a subset of AOE

	Gold Standard	Prompt-All (Cell F1 = 0.0)	Schema-Guided (Cell F1 = 0.64)
Instance: fimmu-17-1806045 — Takayasu Arteritis with IBD (5 rows, 5 target columns)			
Schema	No_references, Predominant_phenotype, Number_of_cases, Treatment, Outcome	LVV_type, IBD_type, Age, Sex, Diagnostic_order, Key_presentation, Treatment, Outcome	No_references, Predominant_phenotype, Number_of_cases, Treatment, Outcome
Col Recall	—	40% (2/5 gold cols)	100% (5/5 gold cols)
Row 1	3 (56), TAK+CD, 2, anti-TNFα, Improved	TAK, CD, 16, M, Simultaneous, Abdominal pain, Prednisone, Asymptomatic	3 (56), TAK+CD, 2, anti-TNFα, Improved
Failure mode	—	Adds 6 unrequested columns; disaggregates summary rows into per-patient entries	Correct schema; minor value differences

Table 12: Case study illustrating schema divergence without guidance. Without schema guidance (Prompt-All), the model constructs a clinically reasonable but evaluation-incompatible schema: it invents 6 columns not in the gold standard and disaggregates summary rows into individual patient entries. Only 2 of 5 gold columns are recoverable after alignment, yielding Cell F1 = 0. With Schema-Guided prompting, the same model produces a perfect schema match (Column Recall = 100%) and achieves Cell F1 = 0.64.

	Gold	Prediction (Claude-Opus-4.6, Schema-Guided)
Intracranial Osteochondroma (5 rows, 6 target columns; Col Recall = 1.0, Row F1 = 1.0, Cell F1 = 0.61)		
Row 1	Location: Middle fossa; Age: 41; Sex: M; Symptoms: Visual disturbance, diplopia; Tx: PSR; Improvement: No	Location: Posterior clinoid; Age: 34; Sex: M; Symptoms: Visual disturbances, headache; Tx: PSR; Improvement: Yes
Row 2	Location: Foramen magnum; Age: 73; Sex: M; Symptoms: Quadriparesis, headache; Tx: TR; Improvement: Yes	Location: Foramen magnum; Age: 73; Sex: M; Symptoms: Quadriparesis, headache; Tx: TR; Improvement: Yes
Row 4	Location: Convexity; Age: 25; Sex: M; Symptoms: Headache, dizziness, speech disorders; Tx: TR; Improvement: N/A	Location: Posterior clinoid; Age: 34; Sex: M; Symptoms: Visual disturbances, headache; Tx: PSR; Improvement: Yes

Table 13: Full medical case study (schema-guided). Row 2 is correctly extracted. Rows 1 and 4 both exhibit **row-level case misalignment**: their predictions are identical, showing the model maps two distinct patients onto the same case description from the literature review. Row 1 appears "partially correct" only because a few fields (Sex, Tx) happen to share values across patients, not because extraction itself partially succeeded. Red = incorrect cells.

tasks. A graduate-level annotator (non-expert in the computer science and medical domains) was asked to extract structured tables from the source documents, following the same schema guidance provided to models. The annotator was permitted to use online search tools (e.g., web browsers) during annotation, simulating a realistic open-book setting.

Setup. We sampled a subset of cases across all four domains. Annotations were evaluated using the same LLM-based evaluation pipeline as model outputs.

Results. Table 14 reports the human annotator's performance compared with representative LLMs.

Analysis. The human annotator substantially outperforms all LLMs on Financial (92.9% vs. best model 46.6%) and Legal (85.8% vs. 50.4%), demonstrating that these tasks are solvable with careful reading. In contrast, the annotator's scores on Academic (42.4%) and Medical (34.0%) are comparable to or below model performance. We attribute this to domain expertise requirements: both domains involve specialized terminology and require familiarity with discipline-specific conventions (e.g., summarizing clinical features from case reports, or extracting methodological details from computer science papers). The annotator, who lacks formal training in these fields, often produced answers that were semantically reasonable but misaligned with the expected granularity or phrasing of the gold standard. This suggests that AOE's Academic and Medical tasks genuinely require domain knowledge beyond general reading comprehension, while Financial and Legal tasks are more accessible to educated non-specialists with sufficient time and access to reference materials.

System	Academic	Legal	Medical	Financial
Human Annotator	42.4	85.8	34.0	**92.9**
GPT-5.4	**48.2**	40.1	14.6	46.0
DeepSeek-V4-Pro	58.6	50.4	16.5	38.7
Claude Opus 4.6	36.3	39.0	21.5	46.6
Gemini 3.1 Pro	45.4	47.9	12.9	40.5

Table 14: Cell F1 (%) comparison between human annotator and LLMs. Model scores are computed on the full test set; human scores are on the sampled subset.

Model	Setting	Academic			Legal			Financial			Avg. Cell F1
		Col F1 (%)	Row F1 (%)	Cell F1 (%)	Col F1 (%)	Row F1 (%)	Cell F1 (%)	Col F1 (%)	Row F1 (%)	Cell F1 (%)	
GPT-5.4	Direct	71.2	63.8	48.2	72.0	71.0	40.1	62.9	91.2	46.3	44.9
	RAG	62.2	38.8	32.7	50.4	44.6	19.6	57.9	92.0	44.5	32.3
Claude Opus 4.6	Direct	71.2	57.1	36.3	70.5	73.1	39.0	60.9	90.3	46.7	40.7
	RAG	67.9	37.1	23.1	51.8	60.6	28.6	57.0	95.7	45.0	32.2
Gemini 3.1 Pro	Direct	71.0	60.4	45.7	69.8	76.3	49.7	51.7	89.6	41.6	45.7
	RAG	59.1	46.8	34.2	54.5	62.7	22.1	50.4	100.0	45.9	34.1
DeepSeek V4-Pro	Direct	71.2	77.5	**58.6**	65.5	80.4	**50.4**	51.8	87.5	39.9	**49.6**
	RAG	59.2	43.6	28.5	41.9	69.8	31.0	48.2	74.8	34.2	31.2
Qwen 3.6-Plus	Direct	70.6	70.4	55.2	65.9	84.4	55.2	52.0	82.5	38.4	49.6
	RAG	67.8	46.1	33.3	49.7	68.3	31.9	51.7	95.4	41.4	35.5

Table 15: Comparison of Direct (full-document) vs. RAG (chunk-based retrieval) across three domains.

Algorithm 1 Simplified Row Matching and Evaluation Algorithm

procedure CALCULATEROWMATCHING($P, G, key_cols, llm_prompt$)
  Input: Predicted table P, Gold table G, Key columns key_cols, LLM prompt llm_prompt
  Output: Precision P_r, Recall R_r, F1 score $F1_r$, Matched pairs

  // Build key mappings
  $P_{keys} \leftarrow \{(i, BuildKey(P[i], key_cols)) \mid i \in P\}$
  $G_{map} \leftarrow \{key \rightarrow [indices] \mid key = BuildKey(G[j], key_cols)\}$
  $matched \leftarrow \emptyset$, $available_G \leftarrow \{0, 1, ..., |G| - 1\}$

  ▷ 1. Exact matching
  for (p_i, p_{key}) in P_{keys} **do**
    if $p_{key} \in G_{map}$ and $G_{map}[p_{key}] \cap available_G \neq \emptyset$ **then**
      $g_j \leftarrow$ pick one from $G_{map}[p_{key}] \cap available_G$
      $matched \leftarrow matched \cup \{(p_i, g_j)\}$
      $available_G \leftarrow available_G \setminus \{g_j\}$
    end if
  end for

  ▷ 2. LLM semantic matching for unmatched rows
  $unmatched_P \leftarrow \{(i, key) \mid (i, key) \in P_{keys}, i \notin matched.p_indices\}$
  $candidate_{keys} \leftarrow \{BuildKey(G[j], key_cols) \mid j \in available_G\}$
  for (p_i, p_{key}) in $unmatched_P$ **do**
    $best_{key} \leftarrow CallLLM(p_{key}, candidate_{keys}, llm_prompt)$
    if $best_{key} \neq$ "$None$" and $best_{key} \in G_{map}$ **then**
      $g_j \leftarrow$ pick one from $G_{map}[best_{key}] \cap available_G$
      if g_j exists **then**
        $matched \leftarrow matched \cup \{(p_i, g_j)\}$
        $available_G \leftarrow available_G \setminus \{g_j\}$
      end if
    end if
  end for

  ▷ 3. Calculate metrics
  $N_m \leftarrow |matched|$, $N_p \leftarrow |P|$, $N_g \leftarrow |G|$
  $P_r \leftarrow N_m/N_p$, $R_r \leftarrow N_m/N_g$
  $F1_r \leftarrow 2P_rR_r/(P_r + R_r)$ (if $P_r + R_r > 0$)
  return $P_r, R_r, F1_r, matched$
end procedure

Task ID	Task Objective	Input Documents	Primary Keys	Output Columns
Aca_0	Identify potential citation relationships between scholarly articles and extract the specific text segment that is being referenced.	Multiple research papers.	["Cited paper title", "Referencing paper title"]	["Referenced content", "Label"]
Aca_1	Enumerate the reported {{Metrics}} of proposed methodologies within a research paper, specifically on the {{Dataset_Name}} dataset.	Multiple research papers.	["paper_name", "method", "result"]	["result", "models_and_settings"]
Fin_0	Conduct an analysis of enterprise scale and growth trends.	Annual company reports from 2020-2023.	["Filename"]	["Reporting Period", "Total Assets (CNY)", "Asset Growth Rate (%)", "Revenue (CNY)", "Revenue Growth Rate (%)", "Total stockholders' equity (CNY)", "Weighted Average Return on Equity (%)"]
Fin_1	Perform a comprehensive assessment of profitability.	Annual company reports from 2020-2023.	["Filename"]	["Reporting Period", "Revenue (CNY)", "Net Profit (CNY)", "Net Profit Margin (%)", "Weighted Average Return on Equity (%)", "Net Profit Growth Rate (%)", "Net cash provided by operating activities (CNY)", "Net Cash Ratio (%)"]
Fin_2	Execute a diagnostic analysis of financial risks.	Annual company reports from 2020-2023.	["Filename"]	["Reporting Period", "Total Liabilities (CNY)", "Liability-to-Asset Ratio (%)", "Operating Cash Flow Growth Rate (%)", "Total Assets (CNY)"]
Fin_3	Analyze the creation of shareholder value.	Annual company reports from 2020-2023.	["Filename"]	["Reporting Period", "Share Capital (Shares)", "Total stockholders' equity (CNY)", "Net Profit (CNY)", "Net Asset per Share (CNY/share)", "Net Profit per Share (CNY/share)", "Weighted Average Return on Equity (%)"]

(Continued on next page)

Task ID	Task Objective	Input Documents	Primary Keys	Output Columns
Fin_4	Benchmark the operating performance for the year {{YEAR}} across different companies within the same industry.	Annual reports from different companies in the same industry for the year {{YEAR}}.	["Filename"]	["Reporting Period", "Revenue (CNY)", "Net Profit (CNY)", "Net cash provided by operating activities (CNY)"]
Fin_5	Analyze the balance sheet structure for the year {{YEAR}} across different companies.	Annual reports from different companies in the same industry for the year {{YEAR}}.	["Filename"]	["Reporting Period", "Total Assets (CNY)", "Liabilities (CNY)", "Total stockholders' equity (CNY)", "Share Capital (Shares)", "Debt to Asset Ratio"]
Fin_6	Assess and benchmark the investment value for the year {{YEAR}} across different companies.	Annual reports from different companies in the same industry for the year {{YEAR}}.	["Filename"]	["Net Profit (CNY)", "Total stockholders' equity (CNY)", "Weighted Average Return on Equity (%)", "Earnings per Share (CNY)", "Equity per Share (CNY)"]
$Legal_0$	Retrieve precise legal provisions relevant to given case descriptions.	Multiple case information documents and related legal statutes.	["Case_name"]	["Relevant_index", "Case_number", "Keywords", "Basic_case_facts"]
$Legal_1$	Extract the final verdict for each defendant from case information.	Multiple case information documents.	["Case_name", "Defendant"]	["Charge", "Sentence_term", "Probation", "Fine", "Confiscation", "Deprivation", "Other_judgments", "Case_number", "Basic_case_facts"]
Med_0	Extract structured clinical attributes from medical case reports. Schema derived from original table caption.	Multiple medical case report papers.	["source_doc"]	Task-dependent (e.g., demographics, diagnosis, treatment, outcome, clinical measurements). Schema guidance provided.

Table 16: Details of Tasks Definition

M Use of AI Assistants

During the development of this work, we used GitHub Copilot for code completion in boilerplate tasks such as data loading scripts, file I/O utilities, and standard evaluation metric computations. All algorithm design, experimental logic, prompt engineering, data annotation, and paper writing were performed entirely by the authors. No AI assistant was used to generate research ideas, interpret results, or draft scientific text.