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

Tianyun Zhong[1,2], Guozhao Mo[1,2], Yanjiang Liu[1,2], Yihan Chen[1,2], Lingdi Kong[1,2], Xuanang Chen[2]*, Yaojie Lu[2], Hongyu Lin[2], Shiwei Ye[1], Shanshan Jiang[3], Bin Dong[3], Le Sun[2], Ben He[1,2], Xianpei Han[2]

[1]University of Chinese Academy of Sciences
[2]Institute of Software, Chinese Academy of Sciences
[3]Ricoh Software Research Center Beijing Co., Ltd.
zhongtianyun2023@iscas.ac.cn

Abstract

Large language models (LLMs) are increasingly expected to extract and synthesize structured information from complex real-world documents, yet they typically produce unstructured, hard-to-verify paragraph-style outputs. We introduce **AOE (Arranged and Organized Extraction)**[1], a bilingual benchmark that evaluates LLMs' ability to distill long-form, multi-document inputs (64k–1.8M tokens) into structured tables. AOE comprises 12 tasks across four domains—academic, financial, legal, and medical—each grounded in genuine analytical needs, with schema guidance provided to isolate extraction quality from formatting variance. We evaluate 12 frontier LLMs (both open-weight and proprietary) under four generation settings of increasing autonomy: direct full-document input, RAG-based retrieval, offline document-browsing agent, and online web-search agent. Even the best-performing model achieves only 42.7% Cell F1, and retrieval-augmented approaches often degrade rather than improve performance, revealing that structured extraction from long contexts remains a fundamental challenge—an **"Area of Effect" (AOE)** that impacts all tested systems.

1 Introduction

Real-world analytical tasks—comparing financial performance across annual reports, summarizing clinical evidence across case studies, aggregating experimental results across papers—demand that information scattered in multiple long documents be distilled into structured, verifiable tables. Large language models (LLMs) are increasingly expected to serve as such automated analysts (Wang et al., 2024b; Bai et al., 2023; Zhang et al., 2024a; Li et al., 2025), yet they often produce paragraph-style answers that are difficult to verify and compare (Figure 1(b)).

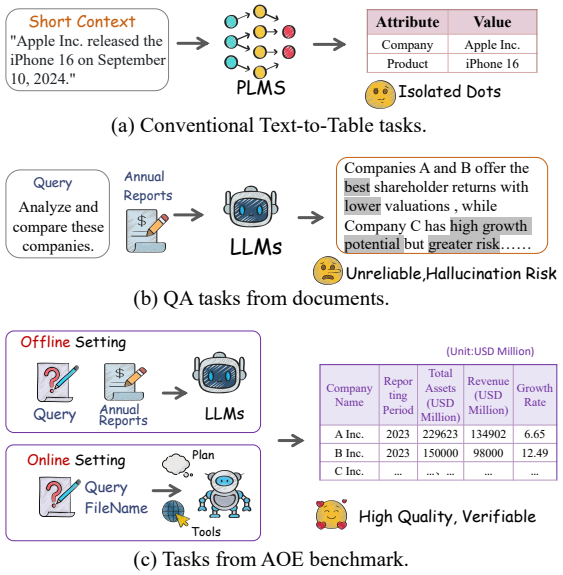


Figure 1: Differences between AOE and previous tasks. Unlike (a) extracting "Isolated Dots" or (b) generating "Unverifiable Output," (c) AOE requires constructing "Organized Evidence" for verifiable analysis.

No existing benchmark adequately evaluates this structured synthesis capability. Long-context benchmarks (Wang et al., 2024b; An et al., 2024) test comprehension without requiring structured outputs; text-to-table benchmarks (Wang et al., 2024a; Deng et al., 2024; Jiang et al., 2024) require structure but are limited to short, single-document inputs; context learning benchmarks (Dou et al., 2026b,a) operate on single documents. None jointly evaluates multi-document reasoning and structured knowledge construction over long contexts—precisely the capability that real-world analysis demands.

To bridge this gap, we introduce **AOE (Arranged and Organized Extraction)**, a bilingual benchmark spanning four domains (academic, financial, legal, and medical) with 2,166 source doc-

*Corresponding author.
[1]Dataset: https://huggingface.co/datasets/tianyumyum/AOE
Code: https://github.com/tianyumyum/AOE

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 3545–3574
November 6-10, 2026 ©2026 Association for Computational Linguistics

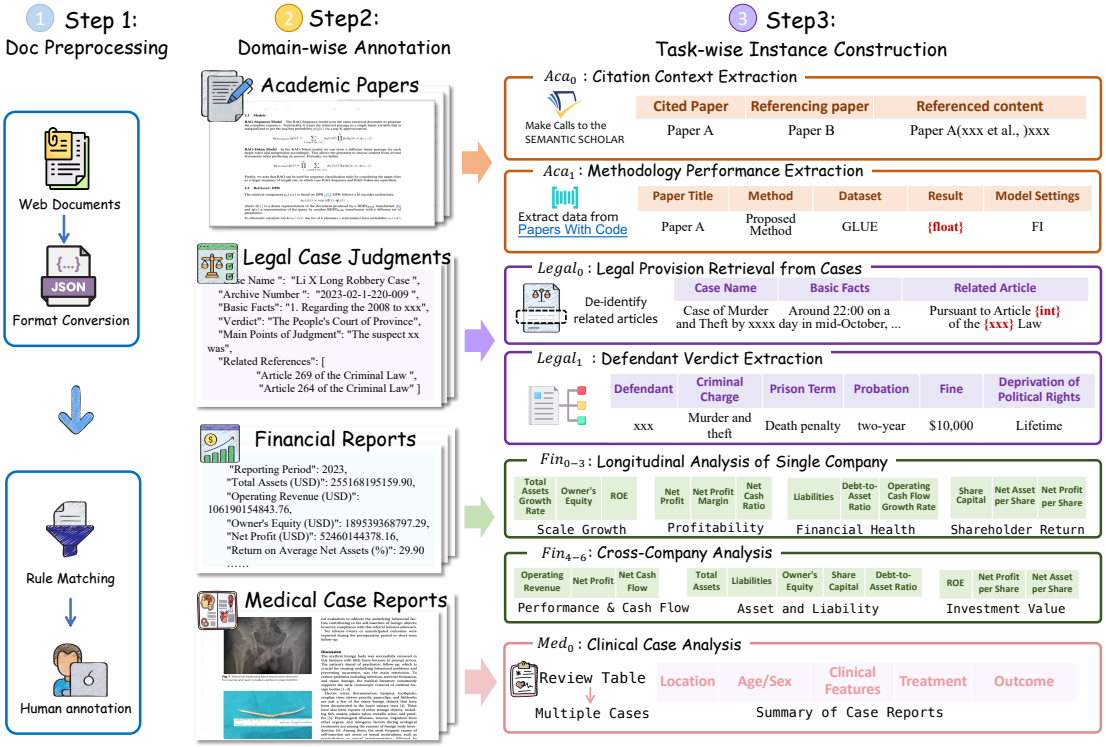


Figure 2: Construction pipeline of AOE Benchmark.

uments and an average input length of 171k tokens (Figure 1(c)). AOE is the first benchmark to jointly evaluate long-context multi-document comprehension and structured table construction, with three key design principles:

Naturally Grounded Tasks: Every task originates from a genuine analytical need, requiring synthesis of up to 13 documents (64k–1.8M tokens) into structured tables.

Hierarchical Evaluation: An automated pipeline assesses structural parsability, overall quality via LLM-evaluator, and cell-level factual accuracy (Cell F1).

Multiple Generation Settings: Beyond direct full-document input, we evaluate RAG, offline agent, and online agent settings to probe different capability axes.

Even the best-performing model achieves only 42.7% *Cell F1*, leaving over half of target facts missed or incorrect and demonstrating a widespread **"Area of Effect" (AOE)** impact across all tested systems. Our contributions are:

- We introduce AOE, the first benchmark that jointly evaluates long-context multi-document comprehension and structured table construction, spanning four domains with 2,166 source documents.
- We design a hierarchical evaluation framework combining structural validation, LLM-based quality scoring, and cell-level factual F1, with verified cross-judge consistency and run-to-run stability.
- We conduct a comprehensive study of twelve frontier LLMs across four generation settings (Direct, RAG, Offline Agent, Online Agent), revealing that current models and retrieval-augmented pipelines remain far from solving real-world structured extraction.

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 20 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-

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	72k	435k	8k	23k
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	6.3/10	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 (chars)	250K	1.8M	64K	104K

Table 2: Data statistics of AOE benchmark.

table tasks are constrained by single-document inputs (Wang et al., 2024a; Deng et al., 2024; Jiang 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.

AOE spans four domains with widely varying input scales (Table 2): Financial tasks average 1.8M tokens, Academic 250K, Medical 104K, and Legal 64K. We further split the Financial domain into **Fin-S** ($\leq$200k tokens, 101 instances) and **Fin-L** ($>$200k tokens, 123 instances) to isolate the effect of context length on extraction performance. Fin-L instances, which often involve multi-year cross-company comparisons, frequently exceed the context windows of current models. However, key financial figures typically appear multiple times across different sections of the same report (e.g., in both the financial highlights and the detailed statements), providing redundant evidence that models can potentially exploit even under truncation. For models with bounded context windows, inputs exceeding the maximum length are truncated.

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[2]. We search for systematic reviews in Q1/Q2

[2]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 (11 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.

journals that contain "summary of reported cases" 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 F.

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 H.

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 G.

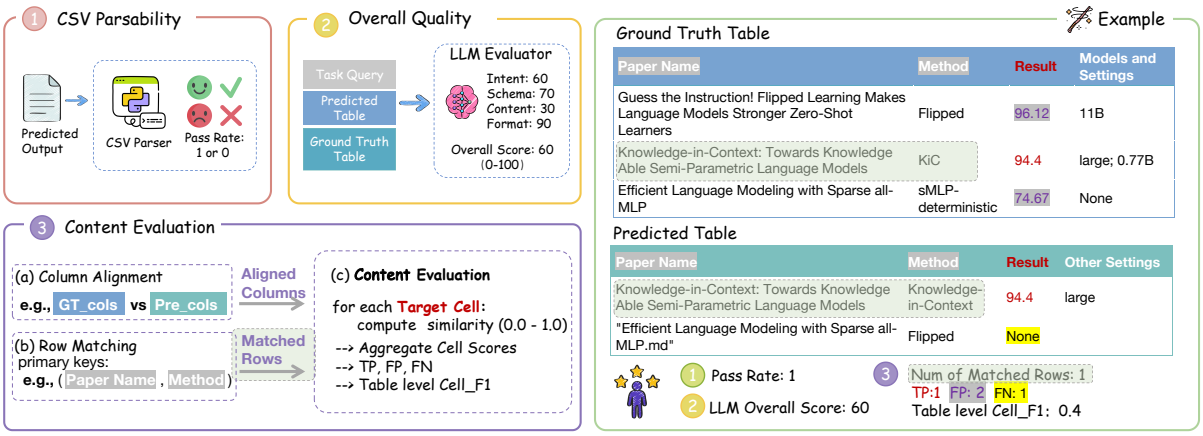


Figure 3: Overview of the Automated Evaluation Pipeline. The pipeline evaluates generated tables from three perspectives: (1) CSV Parsability for basic structural correctness; (2) Overall Quality via an LLM evaluator for a holistic score; and (3) Content Evaluation to calculate a Cell F1 score based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

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 O.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) Cell F1 operates at two levels. At the *cell level*, an LLM evaluator assigns each aligned predicted–gold cell pair a semantic matching score $s_{ij} \in [0, 1]$. This allows cells containing complex or multi-item content to receive partial credit rather than requiring exact string matching. For list or array values, partial matches are scored according to item-level coverage with a penalty factor of 0.8 for incomplete or extraneous content (see Appendix O.2.3 for the detailed scoring rules).

At the table level, we aggregate the cell-level semantic matching scores into precision, recall, and

Model	Ctx. Limit	Pass Rate	Cell F1 by Domain					Overall Evaluation			
			Acad.	Legal	Fin-S	Fin-L	Med.	LLM	Col.	Row	Cell
Open-Weight Models											
DeepSeek-V3.2	131k	92.2	36.9	45.5	57.1	30.1	21.6	74.1	73.7	76.0	38.2
Qwen-3.6-Plus	131k	94.4	52.2	48.2	48.8	22.7	26.0	**75.5**	74.3	78.2	39.6
Kimi-K2.6	131k	96.0	34.5	43.3	**63.4**	**35.3**	23.4	73.8	82.8	75.2	40.0
GLM-5.1	200k	96.2	42.1	44.4	52.9	33.4	**28.9**	74.9	78.0	78.2	40.3
Kimi-K2.5	131k	**98.0**	34.1	50.0	62.5	35.0	28.4	74.0	**83.0**	76.7	**42.0**
DeepSeek-V4-Pro	131k	96.2	**58.6**	**50.4**	52.8	24.6	23.7	72.9	76.1	**82.3**	**42.0**
Proprietary Models											
Gemini-3.1-Pro	131k	87.9	42.8	**44.3**	53.1	27.1	16.1	68.7	68.0	71.6	36.7
Gemini-3-Flash	1M	77.4	44.2	38.3	44.4	**55.1**	6.1	69.6	64.3	61.9	37.6
Claude-Sonnet-4.6	200k	98.7	40.2	36.6	**60.4**	32.7	27.6	75.3	**86.2**	73.4	39.5
GPT-5.4	128k	94.9	**48.2**	40.1	59.9	32.7	18.7	75.3	80.5	76.9	39.9
Claude-Opus-4.6	200k	**99.6**	36.3	39.0	56.0	38.6	**31.4**	**75.4**	83.7	**80.4**	40.3
Seed-2.0-Pro	262k	93.5	45.3	40.0	60.1	40.7	25.6	75.1	80.2	79.1	**42.7**

Table 4: **Main results on AOE.** We report domain-level Cell F1 and overall evaluation scores: a holistic LLM-as-judge score (**LLM**), Column F1 (**Col.**), Row F1 (**Row**), and Cell F1 (**Cell**). Financial instances are split into **Fin-S** ($\leq$200k total input tokens) and **Fin-L** (>200k total input tokens). **Ctx. Limit** denotes the maximum context length allowed in our evaluation pipeline. Models are sorted by overall Cell F1 in ascending order within each category. Bold indicates the best result within each model category and evaluation column. F1 scores and Pass Rate are reported in %; LLM scores are on a 0–100 scale.

F1. Let $\mathcal{A}$ denote the set of aligned predicted–gold cell pairs, with matching scores $s_{ij} \in [0, 1]$. Let N_{pred} and N_{gold} denote the total numbers of cells in the predicted and gold tables, respectively. We compute

$$S = \sum_{(i,j)\in\mathcal{A}} s_{ij}, \tag{1}$$

$$P_{\text{cell}} = \frac{S}{N_{\text{pred}}}, \qquad R_{\text{cell}} = \frac{S}{N_{\text{gold}}}, \tag{2}$$

$$F1_{\text{cell}} = \frac{2P_{\text{cell}}R_{\text{cell}}}{P_{\text{cell}} + R_{\text{cell}}}. \tag{3}$$

Unmatched cells contribute zero to S but remain in their respective denominators, while partially matched cells contribute their semantic matching scores. Given fixed alignments and cell-level scores, table-level aggregation is deterministic.

Thus, the LLM evaluator is used only for semantic matching at the individual cell level, whereas table-level aggregation follows the standard F1 formulation.

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 K.

3 Experiments

3.1 Experimental Setup

Models Under Evaluation. We evaluate twelve frontier LLMs. **Open-weight models** include DeepSeek-V3.2 (DeepSeek-AI et al., 2025), 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** include 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). We use GPT-4o (OpenAI, 2024) as the LLM judge for cell-level scoring.

Generation Settings. Our primary evaluation uses the **Direct** setting, where full source documents are provided for single-pass inference. We additionally evaluate **RAG** (chunk-based retrieval), **Offline Agent** (iterative local document navigation), and **Online Agent** (open-web search) to study different information-access strategies. We compare these settings in Section 3.4, with implementation details in Appendix D.

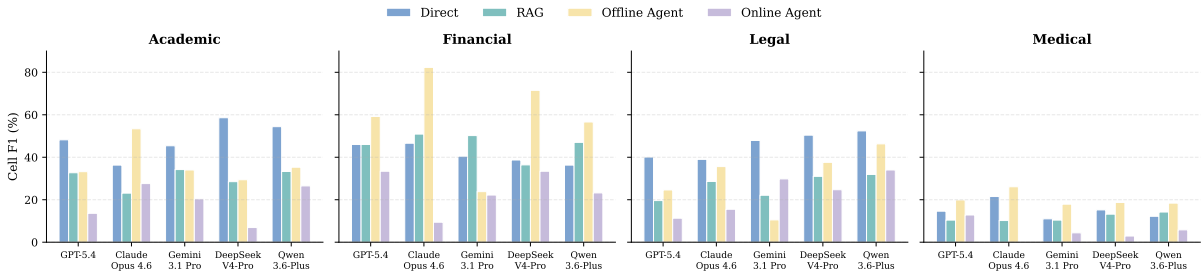


Figure 4: Cell F1 (%) across four generation settings—Direct, RAG, Offline Agent, and Online Agent—for five representative models, broken down by domain.

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 %.

3.2 Main Results: The Bottleneck of Long Contexts

Table 4 shows that AOE remains challenging for all evaluated models: the best performer, Seed-2.0-Pro, achieves only 42.7% average Cell F1, leaving substantial room for improvement.

Similar overall scores mask heterogeneous capability profiles. Despite the narrow range of average Cell F1 scores (36.7–42.7%), performance varies substantially across domains. For example, DeepSeek-V4-Pro achieves the highest Academic Cell F1 (58.6%) but only 24.6% on Fin-L, whereas Gemini-3-Flash reaches 55.1% on Fin-L but 44.2% on Academic. These results suggest that aggregate scores obscure substantial differences in domain-specific capabilities.

Long financial inputs remain particularly challenging. For 11 of the 12 evaluated models, Cell F1 decreases from Fin-S to Fin-L, often by more than 20 points. For example, DeepSeek-V4-Pro drops from 52.8% to 24.6%, and Claude-Sonnet-4.6 from 60.4% to 32.7%. Gemini-3-Flash is the only exception, improving from 44.4% to 55.1%. This pronounced degradation motivates our subsequent analysis of retrieval and agentic approaches for handling long inputs.

Open-weight and proprietary models perform comparably overall. The strongest models from the two groups achieve similar average Cell F1 scores (42.0% for DeepSeek-V4-Pro and 42.7% for Seed-2.0-Pro), with substantial overlap across the leaderboard. Model access type alone therefore does not explain performance differences on AOE.

3.3 Effect of Schema Guidance

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

Table 5 shows that schema guidance substantially reduces schema divergence. Column Recall increases by 21–30 points across all three models, while Cell F1 improves by 3–11 points. The gains are substantially larger for column alignment than for row recovery, suggesting that schema guidance primarily reduces variation in schema construction and allows Cell F1 to more directly reflect content extraction quality. A qualitative case study is provided in Appendix L.

3.4 Can Retrieval and Agents Help?

Beyond the Direct setting, we evaluate whether alternative mechanisms for accessing source information can improve structured table construction (Figure 4). RAG uses FAISS-based chunk retrieval, Offline Agent enables iterative navigation over local documents, and Online Agent provides external search and webpage-reading tools.

Model	Table Construction			Cell Correctness			
	Column	Row	Coverage	Num.	Calc.	Reason.	Text
DeepSeek-V3.2	67.6	**89.8**	50.6	90.0	43.9	41.6	25.7
Kimi-K2.6	75.8	80.5	54.0	85.0	46.8	**44.4**	26.6
Claude-Sonnet-4.6	**77.0**	84.1	**55.8**	86.2	45.7	27.1	**30.2**
Gemini-3-Flash	69.8	86.9	55.7	87.5	**58.5**	31.7	19.9
GPT-5.4	73.8	88.6	54.0	**91.2**	45.1	36.2	19.2
Seed-2.0-Pro	69.9	85.0	53.9	83.8	53.6	43.4	21.8

Table 6: **Error decomposition in the offline setting.** We trace model failures through three stages of table construction: Column recovery (Column), row recovery (Row), and cell coverage (Coverage); then further report Cell F1 by reasoning type: numeric extraction (Num.), financial calculation (Calc.), legal reasoning (Reason.), and textual summarization (Text). Results for all models and raw matched/gold counts are provided in Appendix N.

Retrieval alone does not consistently improve extraction. RAG substantially degrades Cell F1 on Academic and Legal tasks (e.g., DeepSeek-V4-Pro Academic: 58.6% → 30.7%; Qwen-3.6-Plus Legal: 48.2% → 30.9%). In contrast, RAG can benefit some Financial tasks, where source inputs are substantially longer. These results suggest that standard semantic retrieval can alleviate context pressure in some long-input settings but is insufficient for preserving the distributed evidence required by many AOE tasks. A length-stratified analysis is provided in Appendix I.2.

Agentic browsing helps primarily on ultra-long inputs. Offline Agent substantially improves Financial performance for several models, including Claude-Opus-4.6 (46.7% → 60.9%) and DeepSeek-V4-Pro (39.9% → 62.0%), suggesting that selective, iterative document navigation can help when full-document input becomes impractical. However, these gains do not consistently transfer to Academic and Legal tasks, where Offline Agent often underperforms Direct inference. Online Agent performs worst overall across most model–domain combinations, showing that greater autonomy and external information access do not necessarily translate into more accurate structured extraction.

3.5 Where Do Models Fail?

Models recover structure better than content. As shown in Table 6, models recover most target rows (80–90%), yet populate only 45–56% of gold cells. Thus, recovering the table structure does not guarantee complete content extraction: a substantial fraction of the target table remains unanswered even when the relevant rows are identified.

Extraction is easier than reasoning and synthesis. Among populated cells, direct numeric extraction is comparatively reliable (84–91% among the models shown), whereas calculation and legal reasoning are substantially harder, and textual synthesis is the most challenging (15–30%). Together, these results reveal two complementary bottlenecks in AOE: *coverage*, whether models populate the required cells, and *correctness*, whether they correctly transform and synthesize the evidence needed for those cells. Additional breakdowns and qualitative examples are provided in Appendix N.

Human performance reveals different sources of difficulty. On a small 16-instance reference subset, a tool-assisted graduate annotator achieves high Cell F1 on Financial (92.9) and Legal (85.8), but substantially lower scores on Academic (42.4) and Medical (34.0). This contrast suggests that benchmark difficulty reflects different factors across domains. We provide the full human reference study in Appendix M.

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. Oo-

long (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 across four generation settings reveals that even the best model achieves only 42.7% Cell F1, exposing fundamental capability gaps in numerical reasoning and cross-document synthesis. Furthermore, retrieval-augmented and agentic approaches do not consistently improve over direct input, and no single model dominates all domains. AOE establishes a challenging new baseline and underscores the need for future research to build truly reliable knowledge extraction systems.

Limitations

AOE focuses on constructing flat relational tables from multi-document evidence, enabling reproducible cell-level evaluation but not covering more complex structures such as hierarchical or nested tables. Since schema guidance is provided in some settings, AOE also does not fully evaluate open-ended schema induction. Some inputs exceed the context windows of current models, although our RAG and agentic settings provide alternative mechanisms for accessing the document collection. Finally, Cell F1 relies on an LLM evaluator for cell-level semantic matching, which may introduce evaluation variance despite the consistency observed in our robustness experiments.

Acknowledgments

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by the National Key R&D Program of China (2024YFC3308000), the Natural Science Foundation of China (No. 62506354, 62572456 and 62306303) and the Postdoctoral Fellowship Program of CPSF under Grant Number GZC20251041.

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.

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.

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.

A Ethical Considerations

AOE is constructed from publicly available academic, financial, legal, and medical documents. We respect the access and usage conditions of the original data sources and do not introduce additional personally identifiable information beyond that contained in the source documents. For potentially sensitive legal and medical content, we retain only information necessary for benchmark construction and evaluation. Potential risks associated with the use of structured extraction systems in high-stakes domains are discussed in the Limitations section.

License and Intended Use. AOE is released under the Creative Commons Attribution 4.0 International (CC BY 4.0) license for research and evaluation of structured information extraction systems. The original source documents are not relicensed by us and remain subject to their respective licenses

Model	Context Window	Cost / Task
Claude-Opus-4.6	200k	$190.8
GPT-5.4	128k	$74.4
Gemini-3.1-Pro	131k	$60.3
DeepSeek-V4-Pro	131k	$13.2
Qwen-3.6-Plus	131k	$11.8

Table 7: Context windows and estimated per-task API inference costs for representative models.

Domain	GT Cells	>32k	>128k	>256k
Academic	1,037	15.5%	1.7%	0.0%
Financial	6,031	24.4%	19.0%	9.9%
Legal	4,075	0.0%	0.0%	0.0%
Medical	3,283	0.3%	0.0%	0.0%
Overall	**14,426**	**11.4%**	**8.1%**	**5.6%**

Table 8: **Position of ground-truth evidence in the input context.** For each ground-truth cell, we locate the first occurrence of its target content in the concatenated source documents and report the proportion appearing beyond different token positions. Financial tasks exhibit the largest long-context tail.

and terms of use. Users are responsible for complying with the access, use, and redistribution conditions of the original data sources.

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, and Kimi-K2.6) have publicly available architecture details and weights documented in 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, and Seed-2.0-Pro) do not publicly disclose parameter counts.

Inference Hyperparameters. For the Direct and RAG settings, we use `temperature=0`. 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.

Context Windows and Inference Cost. Table 7 reports the context window used in our experiments and the estimated per-task inference cost for representative models. Reporting the effective context window also makes potential truncation effects explicit; we analyze the position of ground-truth evidence relative to these context limits in Appendix C.

C Context Coverage and Truncation Analysis

One potential confounding factor in the Direct setting is input truncation: when the concatenated source documents exceed a model's context window, relevant evidence may fall outside the accessible context. We therefore quantify how much ground-truth evidence occurs at different positions in the concatenated input.

For each ground-truth cell, we locate the first occurrence of its target content in the concatenated source documents and measure its position in the input. Table 8 reports the proportion of ground-truth cells whose first occurrence lies beyond 32k, 128k, and 256k tokens.

Overall, 91.9% of ground-truth cells have their first occurrence within the first 128k tokens, whereas only 8.1% first appear beyond this position. This indicates that truncation alone cannot account for the large remaining performance gap: the best model achieves only 42.7% Cell F1 despite the majority of ground-truth evidence occurring within a typical 128k context window. The effect is highly domain-dependent. In Academic, Legal, and Medical tasks, nearly all ground-truth evidence occurs within 128k tokens, whereas Financial exhibits a substantial long-context tail, with 19.0% of ground-truth cells first appearing beyond 128k tokens and 9.9% beyond 256k.

These results suggest that context truncation is an important factor for the longest Financial tasks, but is unlikely to be the primary explanation for the overall extraction gap. Even when relevant evidence is accessible within the context window, models must still identify, integrate, and map that evidence to the appropriate table cells.

To further reduce dependence on fixed context windows, we additionally evaluate RAG and agentic settings (Appendix I), which access source information selectively rather than relying solely on a single concatenated context. These settings provide complementary evidence about the extent to which retrieval and iterative document access can alleviate long-context constraints.

D 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.

D.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.

D.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[3] (Chen et al., 2024) and indexed using FAISS[4] (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 implemented with LangChain[5] (Chase, 2022).

D.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.

D.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[6] and returns the top-10 search result snippets.
- `read_url(url)`: Fetches and extracts the main text content from a URL via Jina Reader[7].

The agent uses Serper for web search and Jina Reader for webpage content extraction.

The agent iterates through up to 10 search-and-read cycles to gather sufficient information before generating the final CSV. This setting evaluates end-to-end information seeking and structuring when source documents are not locally available.

E Additional Analysis of Generation Settings

E.1 Performance by Input Length

To better understand when retrieval-based methods are beneficial, we stratify instances by source-document length and compare Direct and RAG performance within each group.

The results show that RAG is most competitive on the longest inputs, where retrieval can reduce the amount of irrelevant or truncated context presented to the model. For shorter inputs, however, Direct inference generally performs better, suggesting that retrieval can discard evidence required for cross-document synthesis when the complete context can already be processed directly.

These results help explain the domain-level pattern observed in Section 3.4: retrieval is more useful when context length itself becomes a substantial constraint, but less effective when the task depends on evidence distributed across multiple documents or sections.

F 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.

Academic Domain We sourced documents from Semantic Scholar[8] and Papers With Code[9]. Semantic Scholar provides a vast repository of academic

[3] https://huggingface.co/BAAI/bge-m3
[4] https://github.com/facebookresearch/faiss
[5] https://github.com/langchain-ai/langchain
[6] https://serper.dev/
[7] https://jina.ai/reader/
[8] https://www.semanticscholar.org/
[9] https://paperswithcode.com/

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[10]. 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[11] (providing reference and guiding civil cases) and (2) The China National Database of Laws and Regulations[12] (offering official statutory provisions).

Medical Domain We sourced biomedical case report papers from PubMed Central (PMC) Open Access Subset[13]. 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.

G 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.

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

[10] https://static.cninfo.com.cn/
[11] https://rmfyalk.court.gov.cn
[12] https://flk.npc.gov.cn/index.html
[13] https://pmc.ncbi.nlm.nih.gov/

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.

G.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 9. 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 9: 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.

H 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.

H.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 20 presents the complete query schemas and ground truth columns for all tasks, demonstrating how these principles manifest across domains.

H.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 captures verdict details (defendant name, charge, sentence term) for individual defendants, enabling

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.

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.

I RAG Analysis

I.1 Direct vs. RAG

We first compare RAG with Direct inference across Academic, Legal, and Financial tasks. Table 10 reports results for five representative models.

RAG does not consistently improve structured extraction. Across Academic and Legal tasks, RAG reduces Cell F1 for all five evaluated models, with particularly large drops for DeepSeek-V4-Pro (58.6% → 28.5% on Academic) and Gemini-3.1-Pro (49.7% → 22.1% on Legal). The pattern differs on Financial tasks: RAG yields small improvements for Gemini-3.1-Pro (41.6% → 45.9%) and Qwen-3.6-Plus (38.4% → 41.4%), while slightly degrading the other models.

These results suggest that reducing the amount of input context through retrieval does not by itself improve structured extraction. In particular, retrieval can discard evidence required for tasks that depend on information distributed across multiple passages. The relatively different behavior on Financial tasks motivates a closer analysis of how RAG performance varies with document length.

I.2 Performance by Document Length

To better understand when retrieval-based generation becomes challenging, we construct a standardized evaluation subset of 140 cases, evenly split into **short** (70 cases) and **long** (70 cases) document groups. Table 11 reports the full results.

Across all evaluated models, RAG performs substantially worse on the long subset, with Cell F1 decreasing by 10–17 absolute percentage points. Two factors may contribute to this degradation: (1) **increased retrieval noise**: as document length grows, the retrieval index contains more chunks, making it harder for a fixed top-k retrieval budget to consistently surface the relevant evidence; and (2) **fragmented evidence**: the evidence required for a single cell may span multiple distant passages, while independent chunk-level retrieval may fail to recover all relevant pieces within a single top-k pass. The following case study illustrates the first failure mode, where semantically relevant retrieval fails to surface the factual evidence required by the target table.

Model	Setting	Academic			Legal			Financial			Avg. Cell F1
		Col	Row	Cell	Col	Row	Cell	Col	Row	Cell	
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 10: **Direct vs. RAG performance across domains.** Direct provides the full source documents, whereas RAG provides the top-k retrieved chunks. **Bold** indicates the higher Cell F1 between Direct and RAG for each model–domain pair and for the overall average. Col, Row, and Cell denote Column F1, Row F1, and Cell F1, respectively. All values are reported in %.

Model	Length	Cell F1	Col F1	Row F1
CLAUDE-OPUS-4.6	Short	59.5	62.0	98.5
	Long	41.8	51.7	92.9
DEEPSEEK-V3.2	Short	54.9	57.4	98.5
	Long	37.8	48.9	88.0
GEMINI-3.1-PRO	Short	55.0	53.2	100.0
	Long	43.8	46.6	100.0
GPT-5.4-0305	Short	58.5	62.9	94.6
	Long	41.6	52.6	89.3
QWEN-3.6-PLUS	Short	54.6	56.3	100.0
	Long	38.7	46.7	90.5

Table 11: **RAG performance by source-document length.** We split the evaluation cases evenly into short and long source documents. Across models, longer documents consistently reduce Cell F1 and Column F1, while Row F1 is comparatively more stable. All values are in %.

I.3 Case Study: Semantic vs. Factual Retrieval

Table 12 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 example illustrates a limitation: 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 12: A RAG failure case. RAG retrieves semantically related passages (topical relevance) but misses the numerical evidence (factual precision) required for structured extraction.

J Content Evaluation Details

We provide additional details of the content evaluation procedure described in Section 2.4.2. The overall procedure is summarized in Algorithm 1. Content evaluation consists of two stages: cell-level semantic matching and table-level F1 aggregation.

Cell-level Semantic Matching. For each aligned predicted–gold cell pair, an LLM evaluator assigns a semantic matching score in $[0, 1]$. This soft matching is necessary because a cell may contain complex or multi-item content for which exact string matching would be overly restrictive.

For list or array values, the score reflects item-level coverage with an additional penalty for incomplete or extraneous content. Specifically, we apply a factor of 0.8 to partial list matches. For example, given a gold cell `["Art.14", "Art.39", "Art.40"]` and a prediction `["Art.14", "Art.45"]`, one of the three gold

items is correctly recovered, resulting in a score of $(1/3) \times 0.8 = 0.27$. Detailed evaluator prompts and scoring criteria are provided in Appendix O.2.

Table-level F1 Aggregation. After cell-level scores are fixed, we aggregate them at the table level. The true-positive contribution is the sum of semantic matching scores over correctly aligned cells. Predicted cells absent from the ground-truth table are counted as false positives, while gold cells missing from the prediction are counted as false negatives. Cell precision, recall, and F1 are then computed using the standard F1 formulation.

This two-level design separates semantic evaluation from table-level aggregation: the LLM evaluator is used only to assess semantic similarity within individual cells, while the final Cell F1 is computed deterministically.

K 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).

K.1 Run-to-Run Score Stability

Table 13 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.

K.2 Ranking Stability

Table 14 examines whether model *rankings* are stable across runs—a stricter test than absolute score stability.

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.

K.3 Value Type Stability

Table 15 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 ($\sim$29%), requiring cross-document inference or open-ended synthesis respectively.

L Schema Guidance: Case Study

Table 16 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.

M Human Reference Study

To provide additional context for task difficulty, we conduct a small-scale human reference study on 16 AOE instances spanning all four domains. A graduate-level annotator follows the same schema guidance provided to the models and is allowed to use online search tools during annotation, approximating an open-book extraction setting. The annotator does not have specialized expertise in all evaluated domains, particularly the Academic and Medical tasks. We therefore treat these results as a limited human reference rather than a strict human performance ceiling.

Setup. The human annotations are evaluated using the same cell-level evaluation pipeline as the model outputs. Because the human study covers only a small subset of the benchmark, these results are intended to contextualize domain-level task

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 13: Multi-run performance (Mean $\pm$ 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 14: 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 15: Cell F1 (%) by answer value type, averaged across 5 models $\times$ 5 runs. Low SD confirms metric stability across value categories.

difficulty rather than provide a controlled human–model comparison.

Results and Analysis. The human reference reveals substantial differences in the sources of difficulty across domains. The annotator achieves 92.9 Cell F1 on Financial and 85.8 on Legal. Financial tasks primarily require locating specific figures in very long inputs, which can reach approximately 1.8M characters, while Legal tasks require identifying masked statutes and separating rulings across multiple parties within the same case. The strong human performance suggests that, with sufficient time and access to search tools, many instances in these domains are tractable for a careful human, leaving substantial headroom for current models.

In contrast, the annotator achieves only 42.4 Cell F1 on Academic and 34.0 on Medical. Academic tasks require reading research papers and extracting precise experimental results, while Medical tasks require interpreting and summarizing clinical information from case reports. Given the annotator's limited expertise in these domains, these results suggest that specialized domain knowledge is an important component of their difficulty.

Taken together, these results help distinguish different sources of difficulty in AOE. Financial and Legal tasks remain challenging for models even when a tool-assisted human performs well, whereas Academic and Medical tasks additionally place substantial demands on domain-specific interpretation. This pattern is also consistent with our error decomposition in Appendix N, where textual synthesis is among the most challenging cell categories across models. The human results therefore provide complementary evidence that benchmark difficulty reflects both information-access challenges and domain-specific interpretation, rather than a single source of failure.

	Gold Standard	Raw Prediction (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 16: Case study illustrating schema divergence without guidance. Without schema guidance (Prompt-), 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.

System	Acad.	Legal	Med.	Fin.
Human	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 17: **Human reference and model performance across domains.** Cell F1 (%).

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

N Detailed Error Analysis

Table 18 provides the full error decomposition under the offline setting, together with the raw matched/gold counts behind the aggregate statistics reported in the main text. Across models, row recovery is consistently high (80.2–89.8%), whereas cell coverage is substantially lower (44.9–55.8%). This gap indicates that recovering the overall table structure does not necessarily translate into successfully populating its cells.

The decomposition by cell type reveals a similarly consistent pattern. Models perform best on direct numeric extraction (81.2–96.2%), while cells requiring additional computation or reasoning are more difficult: financial calculation ranges from 34.4–58.5% and legal reasoning from 27.1–46.3%. Textual summarization is the most challenging category, with Cell F1 ranging from only 15.1–30.2%. These results suggest that the main difficulty lies not only in locating the appropriate rows and columns, but also in grounding, transforming, and synthesizing evidence into the correct cell values.

Patient-case misalignment. Beyond missing or incorrectly generated cells, we observe cases in which the model retrieves plausible evidence but associates it with the wrong entity. Table 19 shows a representative example from the Medical domain under the schema-guided setting. Claude-Opus-4.6 recovers all target columns and rows (Col Recall $= 1.0$, Row F1 $= 1.0$), yet achieves only 0.61 Cell F1.

The model correctly extracts Row 2, but populates Rows 1 and 4 with evidence from the same patient case ("Posterior clinoid", age 34). The resulting table is therefore structurally correct while several cell values are assigned to the wrong patients. Some cells in Row 1, such as Sex and treatment, happen to match the gold values because the patients share these attributes, rather than because the correct patient was identified.

This case further illustrates the distinction between *structural recovery* and *content grounding*: correctly recovering the target rows and columns does not guarantee that the extracted evidence is associated with the correct entity.

O Prompt Examples

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

O.1 Prompt Examples for Generating Tables

O.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
    ```
```

	Structure & Coverage			Cell F1 by Content Type			
Model	**Schema**	**Row**	**Coverage**	**Numeric**	**Calculation**	**Reasoning**	**Text**
Open-Weight Models							
DeepSeek-V3.2	67.6	**89.8**	50.6	90.0	43.9	41.6	25.7
Qwen-3.6-Plus	63.3	80.2	44.9	93.8	34.4	45.1	21.9
Kimi-K2.6	75.8	80.5	54.0	85.0	46.8	44.4	26.6
GLM-5.1	64.5	86.1	49.0	**96.2**	41.9	32.3	19.4
Kimi-K2.5	71.9	81.4	51.5	81.2	44.4	**46.3**	20.5
DeepSeek-V4-Pro	62.2	86.4	45.1	**96.2**	37.7	43.2	18.7
Proprietary Models							
Gemini-3.1-Pro	58.4	88.1	45.3	90.0	38.7	42.5	15.1
Claude-Sonnet-4.6	**77.0**	84.1	**55.8**	86.2	45.7	27.1	**30.2**
Claude-Opus-4.6	71.2	84.5	49.2	86.2	45.3	38.4	21.3
Gemini-3-Flash	69.8	86.9	55.7	87.5	**58.5**	31.7	19.9
GPT-5.4	73.8	88.6	54.0	91.2	45.1	36.2	19.2
Seed-2.0-Pro	69.9	85.0	53.9	83.8	53.6	43.4	21.8

Table 18: **Error analysis across construction stages and content types.** We decompose performance into construction success (**Schema**, **Row**, and cell **Coverage**) and Cell F1 across four content types. Models generally recover rows successfully, but coverage drops substantially. Among answered cells, direct numeric extraction is considerably easier than calculation, reasoning, and textual synthesis. All values are percentages; best results in each column are **bolded**.

O.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
    ```
```

Case	Field	Gold	Prediction (Claude-Opus-4.6)
Row 1	Location	Middle fossa	Posterior clinoid
	Age	41	34
	Sex	M	M
	Symptoms	Visual disturbance, diplopia	Visual disturbances, headache
	Treatment	PSR	PSR
	Improvement	No	Yes
Row 2	Location	Foramen magnum	Foramen magnum
	Age	73	73
	Sex	M	M
	Symptoms	Quadriparesis, headache	Quadriparesis, headache
	Treatment	TR	TR
	Improvement	Yes	Yes
Row 4	Location	Convexity	Posterior clinoid
	Age	25	34
	Sex	M	M
	Symptoms	Headache, dizziness, speech disorders	Visual disturbances, headache
	Treatment	TR	PSR
	Improvement	N/A	Yes

Table 19: **Patient-case misalignment despite correct table structure.** Claude-Opus-4.6 recovers all target columns and rows (Col Recall = 1.0, Row F1 = 1.0) but achieves only 0.61 Cell F1. Row 2 is correctly extracted, while Rows 1 and 4 are both populated with evidence from the same patient case ("Posterior clinoid", age 34). Incorrect predictions are highlighted in red.

```
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
      ```
```

O.2 Prompt Examples for Evaluating Tables

O.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>
```

O.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]
      ```
```

O.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>
```

P Use of AI Assistants

LLMs were used as part of the experimental methodology, including as evaluators for assessing model outputs under predefined evaluation criteria. Separately, we used GitHub Copilot for code completion in routine programming tasks and ChatGPT for language polishing and presentation-related assistance during manuscript preparation. All research design, experimental decisions, data annotation, result analysis and interpretation, and scientific conclusions were conducted and verified by the authors.

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 20: Details of Tasks Definition