FINCARDS: Card-Based Analyst Reranking for Financial Document Question Answering

Anonymous ACL submission

Abstract

Financial question answering (QA) over long corporate filings requires evidence to satisfy strict constraints on entities, financial metrics, fiscal periods, and numeric values. However, existing LLM-based rerankers primarily optimize semantic relevance, leading to unstable rankings and opaque decisions on long documents. We propose FINCARDS, a structured reranking framework that reframes financial evidence selection as *constraint satisfaction* under a finance-aware schema. FINCARDS represents filing chunks and questions using aligned schema fields (entities, metrics, periods, and numeric spans), enabling deterministic field-level matching. Evidence is selected via a multi-stage tournament reranking protocol with stability-aware aggregation, producing auditable decision traces. Across two corporate filing QA benchmarks, FINCARDS substantially improves early-rank retrieval quality over both lexical and LLM-based reranking baselines, while significantly reducing ranking variance, without requiring model fine-tuning or unpredictable inference budgets. The code is available at https://anonymous.4open.science/r/Fincards-0414.

1 Introduction

Financial question answering (QA) over corporate filings is often framed as a retrieval problem, but in practice it is a reranking problem under strict financial constraints. Correct evidence must simultaneously match the queried metric, fiscal period, and entity, and often include an explicit numerical value. These signals are sparsely distributed within filings that span hundreds of pages and are heavily interleaved with boilerplate disclosures and recurring statements. As a result, the problem reduces to reliably reranking candidate passages within a single document (Figure 1) so that the top results satisfy all required financial conditions (Chen et al., 2021, 2022; Zhu et al., 2021).

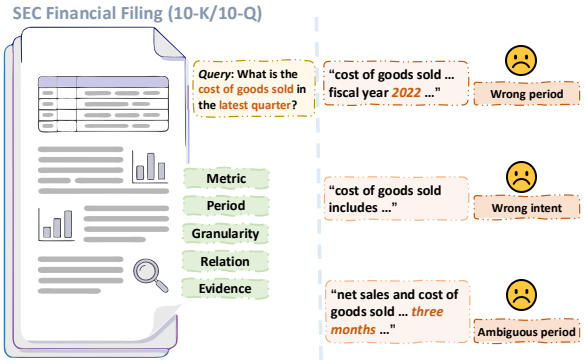

Figure 1: **Key challenge in financial QA.** Reranking must satisfy the correct metric and fiscal period (often numeric), not just semantic relevance. The illustration is based on U.S. SEC corporate financial filings and shows typical failure modes.

A common agent-style approach is to feed large batches of text chunks into a large language model (LLM) and ask it to rank or select evidence. In long corporate filings, this approach breaks down for two practical reasons. First, *scale*: multi-hundred-page reports quickly exceed LLM context budgets, and expanding the input window leads to prohibitive token and latency costs. Second, *opacity*: monolithic, prompt-driven rankings provide little insight into why a particular passage is selected, making the decisions difficult to inspect or audit, which is unacceptable in regulated financial analysis. This leads to systematic errors such as selecting evidence from the wrong fiscal period, misaligning the metric intent of the query, or returning temporally ambiguous passages (Sun et al., 2023; Ma et al., 2023; Choi et al., 2025b). As a result, generic LLM rerankers tend to optimize surface-level semantic relevance rather than explicit constraint satisfaction, making their decisions brittle in financial settings.

Unlike generic LLM rerankers that focus on semantic relevance, FINCARDS aligns evidence using explicit financial fields such as metrics and

periods, producing auditable decision traces. Our design is motivated by how financial analysts reason over long filings: evidence is not evaluated all at once, but progressively narrowed, ordered, and adjudicated under explicit criteria. A junior analyst first screens likely evidence, a senior analyst imposes a coherent global ordering, and a committee resolves close calls when distinctions are subtle.

We operationalize this workflow as a machine-usable alignment contract. Document chunks are abstracted into structured *cards* under a shared schema, questions are mapped to explicit intent specifications over the same fields, and a lightweight tournament-style review performs screening, global ordering, and adjudication to produce a ranked set of evidence passages.

This staged formulation improves numeric and temporal grounding, supports auditability, and keeps computation within predictable cost budgets. Section 3 details the implementation of each stage.

Our focus is the *intra-document* ranking setting: given a single, pre-selected filing, surface the most relevant chunks for a question. This isolates the core retrieval bottleneck in financial QA: locating grounded evidence within long documents before generation (Zhu et al., 2021; Chen et al., 2021).

We make the following contributions:

- We reformulate financial question answering over long corporate filings as an *intra-document evidence reranking* problem under strict numeric, temporal, and entity constraints, shifting the modeling focus away from monolithic long-context reasoning.
- We propose FINCARDS, a structured representation that abstracts document chunks into auditable evidence cards and maps questions into explicit intent specifications, enabling deterministic and interpretable alignment.
- We introduce a zero-shot, tournament-style reranking pipeline that produces stable ranked evidence sets under finance-aware criteria, and demonstrate consistent improvements in early precision over strong baselines.

2 Related Work

Financial QA benchmarks and evidence structure. Early financial QA emphasized numerical reasoning grounded in text and tables. Chen et al. (2021) introduced FinQA with program annotations to make multi-step arithmetic explicit, while Chen et al. (2022) extended this to conversational settings with chained reasoning. Zhu et al. (2021) (TAT-QA) highlighted hybrid text–table reasoning drawn from real reports, making clear that answer correctness depends on metric, period, and entity alignment rather than surface similarity. Subsequent resources (e.g., FinanceBench) found that even strong LLMs misfire in realistic enterprise-style questions, underscoring retrieval bottlenecks and hallucination risks (Islam et al., 2023). More recently, FinDER (Choi et al., 2025a) stressed retrieval-augmented generation in finance with expert triplets that reflect terse practitioner queries, and FinAgentBench (Choi et al., 2025b) isolated *chunk-level* ranking within a selected document, reporting that the paragraph-level setting remains challenging for LLMs and that reinforcement fine-tuning of small models can improve MRR and nDCG. The present work focuses on this intra-document chunk-ranking setting, which remains challenging for LLMs in zero-shot scenarios.

Retrieval and LLM reranking. Classical IR relies on lexical methods such as BM25, while modern pipelines incorporate dense retrievers and cross-encoders (Xiong et al., 2020). Recently, LLMs have been used directly as zero-shot rerankers: listwise approaches like RankGPT and LRL show that instruction-tuned LLMs can reorder candidates competitively without task-specific training (Sun et al., 2023; Ma et al., 2023). However, listwise prompts can be input-order sensitive and context-length constrained. Pairwise prompting (*A vs. B?*) improves calibration and stability (Qin et al., 2024), while setwise/tournament strategies mitigate order sensitivity and scale better with long lists (Zhuang et al., 2024; Chen et al., 2025). Simple rank fusion such as reciprocal rank fusion (RRF) remains a strong baseline to aggregate noisy rankings (Cormack et al., 2009). Recent work suggests that listwise and pairwise comparisons play complementary roles in robust reranking, from global ordering to resolving close decisions. Our work builds on these insights and adapts them to the financial domain by structuring such comparisons within a constraint-driven review process that enforces explicit numeric and temporal alignment.

Agentic reasoning and human review processes. A parallel thread models how humans search and reason. ReAct interleaves reasoning and actions (Yao et al., 2023b), Tree-of-Thoughts explores multiple solution paths (Yao et al., 2023a), Reflexion

adds self-critique with episodic memory (Shinn et al., 2023), and self-consistency improves chain-of-thought reliability (Wang et al., 2023). Work on LLM-as-a-judge surveys reliability and variance when models evaluate content, which is directly relevant to reranking and motivates explicit numeric and temporal guardrails. In finance, documents are visually and numerically dense; layout-aware models (e.g., LayoutLMv3) benefit document QA (Huang et al., 2022), while chart/table QA benchmarks emphasize cross-modal reasoning (Masry et al., 2022; Zhu et al., 2021). Surveys of RAG for LLMs show that most failures are caused by retrieval quality rather than model capacity, highlighting the importance of domain-aligned retrieval (Gao et al., 2024). In this context, our work draws inspiration from agentic and human-aligned reasoning by structuring reranking as a staged review process that emphasizes grounding and explicit constraints, without requiring any model updates.

3 FINCARDS

3.1 Overview

We study *intra-document evidence reranking* for financial question answering: given a user question and all chunks from a single long SEC filing (e.g., 10-K/10-Q), the goal is to rank chunks so that the top-k results satisfy the required financial conditions (metric, fiscal period, entity, and often explicit numbers). Figure 2 summarizes our approach, FINCARDS, which decomposes this problem into three components. **(1) Card abstraction** converts each chunk into a compact, structured *card* that records finance-relevant fields (entities, metrics, periods, numbers, and section cues) for auditable matching. **(2) Query intent mapping** converts the question into a structured intent that specifies the demanded entities/metrics/periods and whether numeric evidence is required. **(3) Tournament reranking** performs staged, zero-shot reranking over cards, combining a screening step, a global listwise ordering step, and a targeted adjudication step, followed by lightweight fusion and post-hoc alignment to produce the final top-k evidence list.

In addition to the final ranking, the pipeline produces an explicit *audit trace* for each selected chunk, recording which Card fields were matched, how the chunk was retained or filtered at each stage, and how its final rank was determined.

3.2 Card Abstraction

Why Card Abstraction. The *card abstraction* module makes financial evidence explicit and auditable before any ranking decisions are made. It converts raw text chunks from long SEC filings into compact, structured records that explicitly encode entities, financial metrics, fiscal periods, and verbatim numeric spans. By operating on these schema fields rather than raw text, downstream stages compare candidates through field-level matching instead of free-form semantic similarity.

By operating on schema fields rather than unconstrained text, downstream stages can perform reliable comparisons under strict numeric and temporal constraints.

Motivation. This design directly addresses recurring failure modes in financial QA over long filings. Lexical retrieval is particularly vulnerable to numeric drift, temporal misalignment, and boilerplate repetition, where legally mandated disclosures dominate surface signals without conveying substantive evidence. Card abstraction mitigates these issues by enforcing explicit temporal normalization, verbatim numeric copying, and boilerplate awareness at the representation level.

Analyst analogy. Conceptually, card abstraction mirrors the preparatory work of a junior financial analyst: relevant numbers, periods, and contextual cues are recorded explicitly so that subsequent reviewers can assess relevance without repeatedly reinterpreting raw text. The resulting Card corpus defines the evidence space for the tournament-style reranking stages described in Section 3.4.

Formal definition. Let $\mathcal{X} = \{x_1, \ldots, x_L\}$ denote the text chunks from a single SEC filing. Each chunk x_i is mapped to a structured *Chunk Card* via a schema-constrained extraction function with deterministic decoding:

$$f : \mathcal{X} \to \mathcal{C}, \qquad c_i = f(x_i) = \mathrm{ChunkCard}(x_i). \tag{1}$$

Each Chunk Card $c_i \in \mathcal{C}$ is a typed record with a full schema, from which we derive a compact *alignment core* for intent matching, while remaining auxiliary fields are used only for screening and stability control:

$$\begin{aligned} c_i &= \left(c_i^{\text{core}}, c_i^{\text{aux}}\right), \\ c_i^{\text{core}} &= (T_i, E_i, M_i, N_i, P_i, S_i, D_i, \Xi_i). \end{aligned} \tag{2}$$

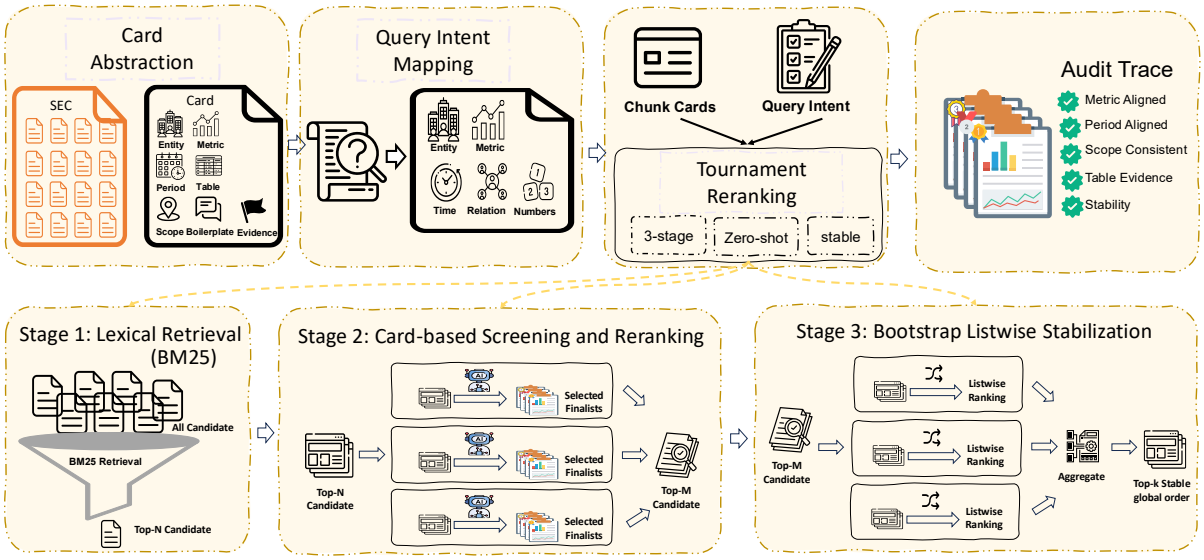

Figure 2: **Overview of the FINCARDS pipeline.** From an SEC filing and a user question, the system constructs structured Cards, a structured query intent, and a tournament reranking module that produces the final Top-k evidence chunks.

The core schema c_i^{core} is an *alignment core (projection)* of the full Chunk Card, used exclusively for intent alignment. Auxiliary fields c_i^{aux} are never used for alignment and are employed only for screening and stability control.

Core fields. $T_i \in \mathcal{T}$ is a topic label; $E_i \subseteq \mathcal{E}$ the set of explicitly mentioned entities; $M_i \subseteq \mathcal{M}$ financial metrics; N_i a (possibly empty) multiset of verbatim numeric spans; $P_i \in \mathcal{P}$ a normalized fiscal period or interval; $S_i \in \mathcal{S}$ the section identifier; $D_i \in \mathcal{D}$ a derived entity–metric–period triple; and Ξ_i evidence spans linking all fields to the original text (*audit trace*).

Auxiliary fields. c_i^{aux} includes derived cues such as scope descriptors, table signatures, and a boilerplate flag. These fields are derived via lightweight rules or parsers over x_i and are used only to guide Stage 2/3 screening and stability control.

3.3 Query Intent Mapping

On the query side, we map each natural-language question to a structured *intent* representation that explicitly encodes entities, metrics, temporal constraints, and numeric requirements. This representation enables direct field-level matching against Card schemas, rather than relying on unconstrained text similarity.

Financial questions are often underspecified in surface form. For example, the query "How did revenue change last quarter?" implicitly requires numeric evidence, a specific fiscal period, and a comparison relation. Intent mapping resolves this ambiguity by decomposing questions into dimensions that can be directly aligned with Card fields.

Formally, each question q_j is mapped to an intent object

$$\text{Intent}(q_j) = \big(T_j, E_j, M_j, R_j, \Theta_j, \nu_j, K_j\big), \quad (3)$$

where T_j is a topic label, E_j entities, M_j metrics, R_j the relation type (e.g., comparison or trend), Θ_j temporal constraints, ν_j whether explicit numeric evidence is required, and K_j lexical keywords.

Together, Card abstraction and intent mapping expose a shared schema for field-level alignment in the tournament reranking stages (Section 3.4).

3.4 Tournament Reranking

We now describe the tournament-style reranking module used to select evidence within a single filing. The pipeline assumes (i) a Card corpus $\mathcal{C}$ derived from filing chunks (Section 3.2), and (ii) a structured query intent $\text{Intent}(q)$ extracted from the question (Section 3.3).

Ranking proceeds in three stages: recall-oriented candidate generation, Card-based semantic filtering, and stability-aware listwise aggregation. This staged design reflects how financial analysts progressively narrow, order, and adjudicate evidence under strict numeric and temporal constraints.

3.4.1 Preliminaries

Given a question q and a filing, let $\mathcal{X} = \{x_1, \ldots, x_L\}$ denote the set of text chunks and $\mathcal{C} = \{c_1, \ldots, c_L\}$ the corresponding Chunk Cards. We extract a structured intent representation $\mathrm{Intent}(q)$ as described in Section 3.3.

The goal of tournament reranking is to return an ordered list of chunk indices $\pi = (\pi_1, \ldots, \pi_k)$ corresponding to the top-k evidence chunks in the filing, optimized for early-rank relevance.

3.4.2 Stage 1: Lexical Retrieval (BM25)

Stage 1 constructs a high-recall candidate set via lexical retrieval within the same filing. Concretely, we score all chunks using the BM25 ranking function (Robertson and Zaragoza, 2009) and keep the top-N candidates:

$$\mathcal{S}_1(q) = \mathrm{TopN}\big(\mathrm{BM25}(q, x_i)\big). \tag{4}$$

To make the candidate budget comparable across filings of different lengths, we use a length-adaptive cutoff

$$N = \mathrm{clamp}(\lceil rL \rceil,\ N_{\min},\ N_{\max}), \tag{5}$$

with $r = 0.5$, $N_{\min} = 60$, and $N_{\max} = 150$ (and $N = L$ if $L < N_{\min}$). Stage 1 serves as a recall-oriented starting point for downstream reranking and highlights cases where relevant evidence receives low lexical scores.

3.4.3 Stage 2: Card-based Screening and Reranking

Stage 2 reduces the Stage 1 candidate set using structured Card representations, without accessing raw chunk text. Starting from $\mathcal{S}_1(q)$, we partition candidates into groups via round-robin assignment so that each group contains a mix of high-, mid-, and low-ranked Stage 1 candidates. Let $\{\mathcal{G}_1, \ldots, \mathcal{G}_m\}$ denote groups of size approximately g (default $g = 25$).

For each group $\mathcal{G}_t$, an LLM agent selects a small set of finalists:

$$\mathcal{F}_t = \textsc{Select}(\mathcal{G}_t, q),\ |\mathcal{F}_t| \in [k_{\min}, k_{\max}]. \tag{6}$$

The selection rubric emphasizes alignment between the query intent and Card fields, including (i) metric overlap, (ii) temporal compatibility, (iii) scope consistency (company-wide vs. segment/region/product), (iv) appropriate evidence type (e.g., table-centric cards for quantitative queries), and (v) down-weighting boilerplate content unless it is uniquely relevant.

To avoid over-filtering, we enforce lightweight coverage constraints. For instance, trend queries must retain at least one temporally grounded, table-bearing card, while definition or policy queries must retain at least one explanatory card.

Group-level selections are merged by union and deduplication, retaining the highest relevance score when a candidate appears multiple times:

$$\mathcal{S}_2(q) = \mathrm{Dedup}\left(\bigcup_{t=1}^{m} \mathcal{F}_t\right). \tag{7}$$

Ties are optionally broken using the original Stage 1 score. If gold retention relative to Stage 1 drops below a threshold, we reshuffle group assignments and repeat selection for a small number of retries.

3.4.4 Stage 3: Bootstrap Listwise Stabilization

While Stage 2 substantially reduces the candidate set using structured Card cues, the resulting ranking can still be unstable due to the sensitivity of single-pass LLM judgments to grouping and comparison context. Stage 3 addresses this issue by enforcing *ranking stability* through multi-round bootstrap listwise aggregation.

Given the Stage 2 candidate set $\mathcal{S}_2(q)$ of size M, we select a group size $g \in [15, 25]$ (adapted to M) and perform up to $R_{\max} = 5$ bootstrap rounds. In each round r, candidates are randomly shuffled with a fixed seed and partitioned into groups $\{\mathcal{H}_{r,1}, \ldots, \mathcal{H}_{r,p_r}\}$. For each group, an LLM produces a complete listwise ranking. Random regrouping exposes each candidate to multiple comparison contexts, mitigating bias introduced by any single partition.

To aggregate rankings across groups and rounds, we employ *normalized Borda scores*. This choice is motivated by two considerations: (i) Borda aggregation preserves fine-grained relative ordering information, rather than relying on hard selection or voting, and (ii) normalization ensures comparability across groups of different sizes, which naturally arise under bootstrap partitioning. For a group $\mathcal{H}$ of size $|\mathcal{H}|$, an item ranked at position ρ receives a score

$$s(\rho; \mathcal{H}) = \frac{|\mathcal{H}| - \rho}{|\mathcal{H}| - 1} \in [0, 1]. \tag{8}$$

Scores are accumulated over all appearances of each candidate:

$$S(i \mid q) = \sum_{r} \sum_{t} s(\rho_{r,t}(i); \mathcal{H}_{r,t}). \tag{9}$$

Sorting candidates by $S(i \mid q)$ yields a stable global ranking π, from which the top-k evidence chunks are returned.

To control computational cost, we monitor convergence of the current top-k set. If the Jaccard similarity between top-k results from consecutive rounds exceeds a threshold (0.9), the procedure terminates early.

3.4.5 Reproducibility and Cost

All LLM interactions use deterministic decoding (temperature $= 0$) and strict JSON schema validation to ensure reproducibility. In addition to the final Top-k ranking, the pipeline outputs an explicit *audit trace* for each selected chunk, recording stage-wise candidate lists, grouping decisions, and per-round ranks in Stage 3. This trace exposes which structured Card fields (e.g., metric, period, scope, and table cues) were matched, how each chunk was retained or filtered across stages, and how its final rank was determined, enabling transparent inspection and controlled multi-model comparisons under identical algorithms and prompts.

In terms of cost, let L be the number of chunks in a filing, N the Stage 1 candidate cutoff, and $M = |\mathcal{S}_2(q)|$ the Stage 2 candidate size. Stage 1 scores all chunks using BM25 in $O(L)$ time per query. Stage 2 requires $m = \lceil N/g \rceil$ LLM calls (one per group), and Stage 3 performs at most $R_{\max}$ bootstrap rounds with $\lceil M/g \rceil$ listwise calls per round. In practice, the total number of LLM calls is often substantially reduced by early stopping once the Top-k ranking stabilizes.

4 Experiments

4.1 Experimental Setup

We evaluate our multi-stage intra-document retrieval and ranking system on the **FinAgentBench** (Choi et al., 2025b) benchmark, which consists of financial question answering tasks derived from U.S. SEC filings (10-K and 10-Q). For each query, the system is provided with a *single financial document* and must identify and rank the most relevant evidence chunks within that document. This setting isolates the challenge of *intra-document retrieval*, where relevant evidence is often sparse, temporally constrained, and interleaved with boilerplate disclosures.

All experiments are conducted in Python 3.x with large language models accessed via API calls. We use deterministic decoding (temperature $= 0$) and structured outputs throughout all experiments.

4.2 System Variants

We compare traditional lexical retrieval, zero-shot LLM-based reranking, and several variants of our multi-stage pipeline. All systems share the same document chunking and evaluation protocol. Throughout this paper, **Stage 1** refers to BM25-based lexical retrieval. For fairness, the zero-shot LLM reranking baseline operates on the same Stage 1 candidate set, with identical candidate budgets.

Stage 1. The lexical retrieval baseline that returns the top-ranked chunks from Stage 1 without any semantic reranking.

Zero-shot LLM Reranking. A standard LLM-based reranking baseline that applies a single-pass, zero-shot LLM to reorder the *Stage 1 candidate set* using raw chunk text. Unlike our approach, this baseline does not leverage structured Card representations, candidate grouping, or multi-round stabilization.

Stage 1 + Stage 3. An ablated variant that applies the Stage 3 bootstrap listwise ranking directly on the Stage 1 candidate set, without Card-based filtering. This setting isolates the effect of stability-aware ranking independent of structured semantic screening.

Stage 1 + Stage 2. A two-stage variant that applies Card-based semantic screening and reranking on top of Stage 1 retrieval, but does not include the bootstrap-based stability mechanism of Stage 3.

Stage 1 + Stage 2 + Stage 3 (Full Pipeline). Our full system, which sequentially combines Stage 1 lexical retrieval, Stage 2 Card-based semantic filtering, and Stage 3 bootstrap listwise stabilization. This progressive design compresses the candidate set while jointly improving ranking accuracy and stability.

4.3 Evaluation Measures

We adopt standard information retrieval metrics at rank 10. We report metrics at $k = 10$ because each question in FinAgentBench is typically associated with a small set of relevant evidence chunks (on the order of ten), making early-rank quality the primary evaluation focus. **nDCG@10** measures graded relevance with emphasis on early ranks, **MAP@10** captures precision across the top-ranked

System	nDCG@10	MAP@10	MRR@10	Cand. Size
Traditional Baseline				
Stage 1	44.26	60.01	69.88	25
LLM-based Baseline				
Zero-shot LLM Reranking	55.80	66.50	78.20	$100 \rightarrow 25$
Ablation of Our Method				
Stage 1+Stage 3	58.23	68.42	77.56	$100 \rightarrow 25$
Stage 1+Stage 2	63.66	73.09	82.95	$100 \rightarrow 40$
Our Full Pipeline				
Stage 1+Stage 2+Stage 3	**71.58**	**78.69**	**89.17**	$100 \xrightarrow{40} 25$

Table 1: **Main results on FinAgentBench under intra-document retrieval.** All retrieval metrics (nDCG@10, MAP@10, MRR@10) are reported as percentages. The proposed three-stage pipeline substantially improves early-rank accuracy while progressively reducing the candidate set size (Cand. Size) at each stage.

results, and **MRR@10** reflects how quickly the first relevant evidence chunk appears. All measures are reported as averages over all evaluation queries with scores multiplied by 100 and reported as percentages.

4.4 Main Results

Table 1 summarizes the main results on FinAgentBench under the intra-document retrieval setting. Our three-stage pipeline achieves a large and consistent improvement over both traditional and LLM-based baselines across all metrics.

Compared to Stage 1, the full pipeline improves nDCG@10 by over **+27 points** and MRR@10 by nearly **+20 points**. Even relative to a strong zero-shot LLM reranking baseline, our approach yields substantial gains (+15.8 nDCG@10), demonstrating that naive LLM reranking is insufficient for financial evidence selection.

Importantly, these accuracy gains are achieved while *progressively reducing the candidate set size* from roughly 100 chunks to fewer than 25, indicating that the proposed design improves both ranking quality and retrieval efficiency.

4.5 Analysis

The results in Table 1 provide clear evidence that each stage of the proposed pipeline contributes meaningfully to retrieval effectiveness.

First, zero-shot LLM reranking improves over BM25 but remains limited, highlighting that replacing lexical scores with unstructured LLM judgments does not adequately resolve temporal mismatch, scope ambiguity, or boilerplate interference in financial filings.

Variant	nDCG@10	MAP@10	MRR@10
Full Card (Baseline)	**63.66**	**73.09**	**82.95**
w/o temporal_data	59.80	69.20	78.50
w/o financial_metrics	61.20	70.85	80.20
w/o tables	58.50	67.80	77.20
w/o scope	62.15	71.80	81.50
Only summary	54.20	63.50	72.80
Raw chunks (no Card)	49.50	60.80	70.15

Table 2: **Stage 2 ablation results.** Each variant removes one component from the Card representation.

Second, introducing Card-based filtering in Stage 2 yields a large performance jump (+7.9 nDCG@10 over zero-shot reranking), confirming that *structured intermediate representations are crucial* for aligning query intent with financial evidence.

Finally, Stage 3 further improves early-rank metrics by stabilizing rankings across multiple comparison contexts. This demonstrates that *ranking stability*, rather than additional semantic filtering alone, is essential for reliable early precision in long, noisy financial documents.

Overall, the analysis validates the core design principles of the proposed system: structured reasoning, progressive filtering, and stability-aware aggregation, all achieved without task-specific fine-tuning.

4.6 Ablation Study

4.6.1 Stage 2: Card Component Ablation

We ablate individual components of the Card representation used in Stage 2, while keeping the pipeline and model fixed.

Table 2 shows that structured Card fields are essential for effective semantic filtering. Removing temporal information causes the largest degradation, highlighting the importance of temporal alignment in financial QA. Ablating table indicators or financial metrics also leads to substantial drops, indicating that identifying quantitative evidence is critical even without exposing raw numbers. Using only free-text summaries performs poorly, and operating directly on raw chunks yields the worst results. Overall, the Card abstraction is a necessary intermediate representation rather than a mere efficiency optimization.

4.6.2 Stage 3: Ranking Strategy Ablation

We further analyze Stage 3 by ablating key design choices in the bootstrap-based listwise ranking

Variant	nDCG@10	MAP@10	MRR@10	Rank Var.
Bootstrap (R=3–5)	**71.58**	**78.69**	**89.17**	0.0342
Single Round (R=1)	68.20	75.20	85.80	0.0856
Fixed Grouping	69.15	76.35	86.95	0.0621
Mean Rank Aggregation	69.80	77.05	87.50	0.0498
Voting Aggregation	67.35	74.80	85.10	0.0723
No Early Stopping	71.65	78.75	89.25	0.0318

Table 3: **Stage 3 ablation results.** Rank variance measures ranking stability across bootstrap rounds.

Stage / Model	nDCG@10	MAP@10	MRR@10
Stage 1 (BM25)	44.26	60.01	69.88
Stage 2: Card-based Filtering			
GPT-5 Mini	63.66	73.09	82.95
GPT-4 Mini	64.63	75.63	84.66
Claude-4.5-Opus	**70.89**	**81.03**	**89.67**
Claude-4.5-Sonnet	67.15	77.16	85.71
Gemini 2.5 Flash	63.28	73.32	82.46
Gemini 3 Pro (Preview)	67.06	76.53	85.98
Stage 3: Bootstrap Stable Ranking			
GPT-5 Mini	71.58	78.69	89.17
GPT-4 Mini	71.15	77.77	87.98
Claude-4.5-Opus	75.72	81.20	89.94
Claude-4.5-Sonnet	74.42	79.88	90.46
Gemini 2.5 Flash	74.87	79.46	89.96
Gemini 3 Pro (Preview)	**76.52**	**81.39**	**91.76**

Table 4: **Robustness across LLM backbones on FinAgentBench.**

procedure. In addition to ranking quality, we report *rank variance* as a stability metric, computed across bootstrap rounds.

Table 3 shows that bootstrap-based aggregation is crucial for both accuracy and stability. Single-round ranking exhibits much higher variance, confirming the instability of one-shot LLM judgments. Random regrouping consistently outperforms fixed grouping, indicating that exposure to diverse comparison contexts reduces bias. Normalized Borda aggregation outperforms simpler aggregation schemes, suggesting that fine-grained relative ordering is important. Disabling early stopping yields only marginal gains while increasing computation, as rankings typically converge within 3–4 rounds. Overall, Stage 3 acts as a stability control layer that improves early precision while controlling variance.

4.7 Robustness Across LLM Backbones

To test model dependence, we run Stage 2 and Stage 3 with six different LLM backbones. All other components of the pipeline remain unchanged.

Analysis. Table 4 shows that the proposed pipeline is robust across diverse LLM backbones.

First, all models exhibit a substantial improvement from Stage 1 to Stage 2, indicating that Card-based semantic filtering consistently improves evidence selection regardless of model capacity. This suggests that the gains are primarily driven by the structured intermediate representation rather than backbone-specific reasoning ability.

Second, Stage 3 further improves or stabilizes ranking quality for every model, with consistent gains in nDCG@10 and MRR@10. This confirms that bootstrap-based listwise aggregation effectively mitigates the variance of single-pass LLM rankings across architectures.

Finally, while stronger backbones achieve higher absolute scores, the relative improvements introduced by Stage 2 and Stage 3 remain similar across model families. This demonstrates that the proposed framework is model-agnostic and complements advances in base LLM capability, rather than relying on them.

5 Conclusions and Future Work

We introduced FINCARDS, a tournament-style, zero-shot intra-document reranking framework for financial QA over long SEC filings. The key idea is to replace monolithic relevance ranking with structured evidence selection, combining Card-based abstractions and a staged reranking protocol that enforces metric, temporal, and scope constraints.

Across extensive experiments on FinAgentBench, FINCARDS consistently outperforms both lexical baselines and strong zero-shot LLM rerankers, while progressively reducing the candidate set size. These gains are achieved without task-specific fine-tuning, demonstrating that reliability in financial QA can be substantially improved through structured intermediate representations and stability-aware decision procedures.

More broadly, our results indicate that reliable zero-shot reranking in financial QA depends on explicit intermediate structure rather than single-pass relevance judgments.

Future work will explore adaptive budget control across stages, cross-document evidence selection, and structured Card interfaces for downstream answer generation.

Limitations

Despite its strong empirical performance, the proposed framework has several limitations.

First, the multi-stage pipeline incurs non-trivial computational cost. In particular, the listwise and bootstrap-based ranking stages require multiple LLM calls per query, which may limit scalability in large-scale or latency-sensitive deployments. While candidate compression and early stopping mitigate this cost in practice, efficiency remains an important consideration.

Second, the current study focuses exclusively on intra-document retrieval within a single SEC filing. Many real-world financial analysis tasks require reasoning across multiple documents or heterogeneous sources, such as press releases and earnings calls. The effectiveness of the tournament-style design in such multi-document settings has not yet been evaluated.

Third, although the approach avoids task-specific fine-tuning, it may still be sensitive to prompt design and schema choices. While we employ strict structured outputs and deterministic decoding to improve stability, further work is needed to understand robustness under prompt variation and evolving model behaviors.

Ethical Considerations

This work studies retrieval and reranking methods for financial question answering over publicly available regulatory filings. The proposed framework operates solely on textual disclosures released by companies and does not involve personal data, user profiling, or sensitive individual information.

The primary goal of this research is to improve evidence selection and interpretability in financial analysis and decision support. By emphasizing structured intermediate representations and transparent ranking procedures, the approach aims to support more responsible use of large language models in high-stakes financial settings.

Potential risks include over-reliance on automated systems and misinterpretation of retrieved evidence if used without appropriate human oversight. Accordingly, the proposed system is intended as an assistive tool for analysts, rather than a replacement for professional judgment. Beyond risks commonly associated with automated information retrieval systems, we do not anticipate significant negative societal impact.

Data License All experiments in this work are conducted on publicly available datasets derived from U.S. SEC filings. The underlying documents, such as 10-K and 10-Q reports, are released under public disclosure requirements and are freely accessible for research purposes.

The FinAgentBench benchmark used in our experiments follows the original data collection and usage terms specified by its authors. This work does not redistribute the original filings, nor does it impose additional licensing constraints beyond those associated with the source datasets.

References

Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. 2025. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. *Preprint*, arXiv:2406.11678.

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

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025a. *FinDER: Financial Dataset for Question Answering and Evaluating Retrieval-Augmented Generation*, page 638–646. Association for Computing Machinery, New York, NY, USA.

Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, and Yongjae Lee. 2025b. *FinAgentBench: A Benchmark Dataset for Agentic Retrieval in Financial Question Answering*, page 632–637. Association for Computing Machinery, New York, NY, USA.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New

York, NY, USA. Association for Computing Machinery.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *Preprint*, arXiv:2311.11944. ArXiv preprint.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *Preprint*, arXiv:2305.02156.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

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

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *Preprint*, arXiv:2007.00808.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 38–47, New York, NY, USA. Association for Computing Machinery.

A Candidate Set and Evaluation Fairness

All reranking-based systems operate on the same Stage 1 (BM25) candidate pool. Specifically, for each query, we retrieve the top-N chunks using BM25 ($N \in [60, 150]$ depending on document length). Zero-shot LLM reranking, Stage 2 filtering, and Stage 3 ranking all receive identical candidate sets at their respective entry points. This design ensures that all reported improvements arise from reranking quality rather than candidate recall advantages or budget discrepancies. In particular, no method is allowed to retrieve additional candidates beyond the Stage 1 pool.

B Prompts and Templates

This appendix reports the prompts used to instantiate the structured representations in FINCARDS. All prompts enforce strict JSON-only input and output schemas to ensure deterministic behavior and reproducibility. We stress that FINCARDS does not rely on prompt wording or prompt-specific heuristics; instead, the prompts serve solely to instantiate fixed interfaces defined by the method.

B.1 Card Abstraction Prompt

The card abstraction prompt converts each document chunk into a structured *Chunk Card*. The card explicitly encodes the chunk's evidence role, temporal anchoring, scope, and verifiability signals, and serves as the primary alignment interface for downstream reranking stages. An excerpt of the prompt used for this abstraction is shown in Figure 3.

System: Financial document analysis expert.

Task: Given a document chunk extracted from a 10-K filing, generate a structured *Chunk Card* that specifies under what conditions the chunk can serve as evidence for reranking.

Output Format (JSON only):

- Identity: chunk_id, section_path, chunk_index
- claim_role (primary_evidence, supporting_context, definition, caveat, boilerplate, structural_heading)
- evidence_type (table_numeric, narrative_numeric, qualitative_explanation, policy_text, guidance, none)
- answerability_profile (single_fact, comparison, trend, aggregation, attribution)
- temporal_anchor (quality and normalized span)
- scope_signature (entity scope, geography, product)
- measurement_basis (GAAP, non-GAAP, adjusted, reported, unknown)
- verifiability (numeric claims, table presence, comparison cues)
- risk_signals (boilerplate likelihood, limitations)
- semantic_sketch (one-sentence claim summary and topic anchors)

Constraints:

- Output must be valid JSON.
- Structural headings have zero evidence capability.
- Temporal fields use YYYY-MM or null.
- Use unknown if scope or measurement basis is uncertain.

Figure 3: Excerpt of the prompt used to instantiate a **Chunk Card (full schema)**.

B.2 Query Intent Mapping Prompt

The query intent mapping prompt converts a natural-language financial question into a structured *Query Intent*. This representation explicitly encodes the topical focus, requested financial metrics, temporal constraints, and relational form of the question, and defines the demand-side requirements used for alignment with Chunk Cards during reranking. An excerpt of the corresponding prompt is shown in Figure 4.

B.3 Stage 2: Batch Selection Prompt

Stage 2 performs group-wise evidence selection within the candidate pool retrieved by Stage 1. Given a user question and a small group of candidate chunks with their corresponding Chunk Cards, the model selects a bounded number of relevant chunks based solely on card-level information. This stage does not rely on external retrieval scores and serves to filter and structure the candidate set before stability-oriented reranking in Stage 3. An excerpt of the batch selection prompt is shown in Figure 5.

B.4 Stage 3: Listwise Ranking Prompt

Stage 3 performs listwise reranking over the filtered candidate sets produced by Stage 2. Given a user question and a group of candidate chunks represented only by their Chunk Cards, the model produces a complete relative ordering from most relevant to least relevant. This stage explicitly avoids numerical calculation and absolute scoring, and is designed to provide stable ordinal judgments that can be aggregated across multiple rounds. An excerpt of the listwise ranking prompt is shown in

System: Financial QA intent extractor for questions over SEC filings.

Task: Given a user question, produce a structured *Query Intent* that captures the information need of the question.

Output Format (JSON only):

- topic (e.g., Revenue, Costs/Expenses, Profitability, Liquidity, Guidance/Outlook, Risk)
- entities (explicitly mentioned companies or segments, if any)
- metrics (requested financial metrics)
- temporal_scope (type, normalized periods, granularity)
- requires_numeric_evidence (true / false)
- relation (lookup, trend, comparison, explanation, definition, policy)
- keywords (salient lexical cues)

Constraints:

- Output must be valid JSON.
- If uncertain, use conservative defaults (e.g., Other, none, or []).
- Temporal expressions should prefer fiscal normalization (e.g., FY2023, latest quarter).

Figure 4: Excerpt of the prompt used to instantiate Query Intents.

Figure 6.

C Error Analysis and Failure Modes

C.1 Methodology

To complement the aggregate retrieval metrics, we conduct a structured error study to analyze the behavior of the proposed pipeline under different failure conditions. Rather than relying on anecdotal examples, our analysis is grounded in *stage-wise retrieval traces* collected from the full system, enabling verification and reproducibility.

We first select a small but representative set of six queries, covering both successful and failed retrieval scenarios. Specifically, the selected cases include: (i) queries where BM25 fails to retrieve any gold evidence but subsequent stages recover relevant chunks; (ii) queries where Card-based alignment remains ineffective; and (iii) queries where bootstrap-based aggregation introduces performance degradation. This selection strategy ensures coverage of all major pipeline components.

For each selected query, we analyze retrieval outcomes at all three stages. At Stage 1, we inspect the full BM25 ranking over the document and record gold chunk statistics, including the total number of gold chunks, their absolute BM25 ranks and scores, and whether they appear in the dynamic Top-N candidate set. This allows us to distinguish between recall failures and ranking failures at the lexical retrieval level.

At Stage 2, we examine the final candidate set produced by Card-based filtering and reranking. For top-ranked candidates as well as representative hard negatives, we extract structured Card attributes, including matched financial metrics, temporal information, table presence, and scope alignment. We additionally record the selection rationale generated by the Stage 2 agent, enabling direct attribution of ranking decisions to specific Card fields.

At Stage 3, we analyze the stability of bootstrap-based reranking for cases where the final ranking differs from Stage 2 or is used to demonstrate convergence. We log per-round top-K candidate sets, early stopping behavior, and aggregation statistics such as rank variance, top-5 frequency, and Borda score accumulation. These signals allow us to identify whether performance changes arise from instability across randomized grouping contexts or from systematic evidence reweighting.

Finally, each case is annotated with a small set of failure-type labels (e.g., lexical mismatch, temporal misalignment, schema coverage gap, or implicit reasoning) and a coarse query intent category (quantitative lookup, trend/comparison, or qualitative impact). This taxonomy enables cross-case comparison and clarifies which error patterns are addressed by the proposed design and which remain open challenges.

Together, this error study provides a fine-grained, auditable analysis of the pipeline, explaining not only *whether* the method succeeds or fails, but also *why* these outcomes occur at different stages.

C.2 Representative Cases

Table 5 presents six representative queries used to illustrate typical success and failure patterns across different stages of the pipeline.

C.3 Findings

Our error analysis yields several consistent and instructive findings about the behavior of multi-stage retrieval under financial QA settings.

First, lexical retrieval failures dominate early-stage errors. Across multiple cases, Stage 1 BM25 fails to retrieve any gold evidence within

the dynamic Top-N candidate set, resulting in zero nDCG@10. These failures are primarily caused by lexical and semantic mismatches, including abbreviations (e.g., "SG&A" vs. "selling, general and administrative expenses"), paraphrased financial concepts (e.g., "cash from operations" vs. "net cash provided by operating activities"), and implicit temporal constraints (e.g., "latest quarter"). This confirms that term-based retrieval alone is insufficient for financial documents, where equivalent concepts are frequently expressed using heterogeneous terminology.

Second, Card-based alignment effectively recovers gold evidence when the query intent is structurally expressible. For quantitative lookup queries involving explicit financial metrics and time scopes, Stage 2 substantially improves retrieval quality. In these cases, gold chunks are consistently promoted due to matched financial metrics, explicit temporal annotations, and the presence of structured tables. Notably, these improvements occur even when Stage 1 recall is zero, demonstrating that Card-based reasoning can compensate for lexical failures by leveraging schema-level alignment rather than surface text overlap.

Third, Card-based methods degrade gracefully but predictably under implicit or explanatory queries. For qualitative or impact-oriented questions (e.g., interest rate implications or strategic decision-making), both Stage 2 and Stage 3 exhibit limited gains and, in some cases, performance degradation. Error traces indicate that such failures arise not from ranking instability but from schema coverage gaps: the Card representation lacks fields to encode implicit reasoning, causal effects, or cross-section narrative synthesis. As a result, the model over-selects superficially related but ultimately non-answering chunks, revealing a fundamental limitation of schema-driven filtering for abstract reasoning tasks.

Fourth, bootstrap-based aggregation improves ranking stability but cannot correct systematic misalignment. Stage 3 bootstrap reranking consistently reduces rank variance and yields highly stable Top-K sets when Stage 2 candidates are well-aligned with the query. However, when Stage 2 admits structurally mismatched candidates, bootstrap aggregation reinforces these errors rather than correcting them. This indicates that Stage 3 primarily serves as a stabilizer rather than a semantic repair mechanism.

Overall, the error study demonstrates that the proposed pipeline is highly effective when query intent can be decomposed into explicit schema-aligned constraints (metric, time, scope, and evidence type). Conversely, failures predominantly arise from intent types that exceed the expressive capacity of the current Card schema, rather than from ranking noise or stochasticity.

System: Financial document analysis expert.

Task: Given a user question and a group of candidate document chunks, select the most relevant evidence chunks to answer the question. Selection must be based exclusively on the provided Chunk Cards.

Inputs:

- Question text
- A group of candidate chunks with their Chunk Cards
- A required selection range ($k_{\min}$ to $k_{\max}$)
- Optional coverage quotas (hard constraints)

Selection Criteria:

- Metric matching between the question and chunk content
- Temporal alignment with the question requirements
- Scope consistency (entity, segment, or region)
- Content type suitability (table vs. narrative)
- Overall relevance inferred from card summaries and signals

Table Handling Warning: Table-based chunks require additional verification of structure, headers, temporal coverage, and metric relevance. Sequential or continuous tables must be interpreted cautiously and should not be selected solely due to the presence of tabular data.
Output Format (JSON only):

- selected_chunks: an ordered list of between $k_{\min}$ and $k_{\max}$ chunks
- For each chunk: chunk_id, selection_reasons, relevance_score (0–100)

Constraints:

- Selection must satisfy mandatory coverage quotas, if provided.
- Chunks are evaluated purely on card information.
- Results are ordered by descending relevance.

Figure 5: Excerpt of the Stage 2 batch selection prompt used for group-wise evidence filtering.

System: Financial document analysis expert.

Task: Given a user question and a group of candidate evidence chunks, rank all chunks from most relevant to least relevant for answering the question. Ranking must rely exclusively on the provided Chunk Cards.

Inputs:

- Question text
- A group of candidate chunks with their Chunk Cards

Ranking Criteria:

- Metric matching between the question and chunk content
- Temporal alignment with the question requirements
- Scope consistency (entity, segment, or region)
- Content type suitability (table vs. narrative)
- Overall relevance inferred from card summaries and signals

Critical Constraints:

- Do not access the original chunk text.
- Do not perform numerical calculations.
- Do not assign absolute relevance scores.
- Rank *all* chunks in the group using relative ordering only.

Output Format (JSON only):

- ranked_chunks: a complete ordered list of all chunks
- For each chunk: chunk_id, rank (1 = most relevant), and a brief reason

Figure 6: Excerpt of the Stage 3 listwise ranking prompt used for stability-oriented reranking.

Case ID	Query Type	Stage 1	Stage 2	Stage 3	Failure / Success Mode	Key Evidence from Trace
qdfaa5e169d37	Quantitative lookup (SG&A, latest quarter)	Fail	Recover	Stable	Lexical + temporal mismatch fixed by Card	Gold ranked >60 by BM25; Card matches `financial_metrics=SG&A` and quarterly temporal data
q2a3f6f1492e8	Quantitative lookup (GAAP operating expense)	Fail	Recover	Stable	Abbreviation mismatch fixed by Card	BM25 misses "GAAP OPEX"; Card aligns metric + income-statement table
q83bb50eb29cd	Qualitative risk disclosure (cybersecurity)	Fail	Strong	Degrade	Bootstrap aggregation instability	Stage 2 near-perfect ranking; Stage 3 shows high rank variance across bootstrap rounds
q4652caf2531b	Quantitative + geographic provenance	Fail	Fail	Fail	Schema coverage gap	Gold evidence requires geography; no Card field expresses origin outside U.S.
q7c5dd2e1ba56	Quantitative lookup (revenue, latest quarter)	Fail	Fail	Fail	Temporal aggregation gap	Query requires cross-quarter aggregation; Card lacks temporal trend encoding
q80a13d0df306	Qualitative strategy / impact	Partial	Partial	Fail	Implicit reasoning beyond schema	Relevant evidence dispersed across narrative sections; no localized Card alignment

Table 5: **Error study summary across representative cases.** Stage-level outcomes are annotated as Recover/Stable, Partial/Degrade, or Fail. The table highlights how Card-based alignment resolves lexical and temporal mismatches, while remaining failures concentrate in schema coverage gaps and implicit reasoning queries.