Human-Aligned Multi-Agent Reranking for Financial Document QA in Zero-Shot Settings

Yixi Zhou
ShanghaiTech University
Shanghai, China
zhouyx2022@shanghaitech.edu.cn

Jiayi Yin
University of Wisconsin-Madison
Madison, WI, USA
jyin66@wisc.edu

Zhongyang Liu
ShanghaiTech University
Shanghai, China
liuzhy12024@shanghaitech.edu.cn

Ruoxin Huang
ShanghaiTech University
Shanghai, China
huangrx2022@shanghaitech.edu.cn

Haipeng Zhang*
ShanghaiTech University
Shanghai, China
zhanghp@shanghaitech.edu.cn

ABSTRACT

Answering financial questions over long corporate reports (e.g., annual and quarterly reports) requires retrieving passages that contain the *correct metric*, *fiscal period*, *entity*, and often *explicit numbers*. Common agent-style approaches that dump large batches of chunks into LLMs face two failures in this setting: they exceed context and compute limits on lengthy filings, and they yield opaque rankings that are difficult to audit. We argue for a *human-aligned* paradigm that treats retrieval as a transparent review process rather than a monolithic prompt. Evidence chunks are first converted into structured cards with entities, metrics, periods, and verbatim spans, while questions are clarified into intents that specify required evidence. A lightweight, analyst-style workflow then follows: an initial screen narrows candidates, a global review imposes a coherent order, and targeted tie-breaks resolve close calls under explicit, finance-aware criteria. The outcome is a zero-shot reranking pipeline that mirrors how analysts search and justify evidence: it keeps token and latency costs predictable, exposes traceable rationales for each selection, reduces temporal and numeric mismatches, and yields stable top-k results suitable for high-stakes financial QA without any task-specific fine-tuning.

CCS CONCEPTS

• **Information systems → Language models**.

KEYWORDS

Financial QA, Chunk Ranking, Zero-shot, Multi-agent, Reranking, Retrieval, FinAgentBench

ACM Reference Format:
Yixi Zhou, Jiayi Yin, Zhongyang Liu, Ruoxin Huang, and Haipeng Zhang. 2025. Human-Aligned Multi-Agent Reranking for Financial Document QA in Zero-Shot Settings. In *Proceedings of The 2nd Workshop on LLMs and Generative AI for Finance (AI4F '25).* ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Financial question answering (QA) over long corporate financial reports (e.g., annual and quarterly reports commonly filed with regulators) is a high-stakes retrieval problem. Correct answers are rarely about vague topical similarity; they hinge on locating passages that contain the *right metric* (e.g., revenue, margin), for the *right fiscal period* (e.g., FY2023 Q4), tied to the *right entity*, and often an *explicit number*. In realistic filings, these signals are interleaved with boilerplate language, forward-looking statements, and repeated disclosures, making *intra-document* retrieval, finding the right chunk inside a single, very long report, both critical and difficult [Chen et al. 2021, 2022; Zhu et al. 2021].

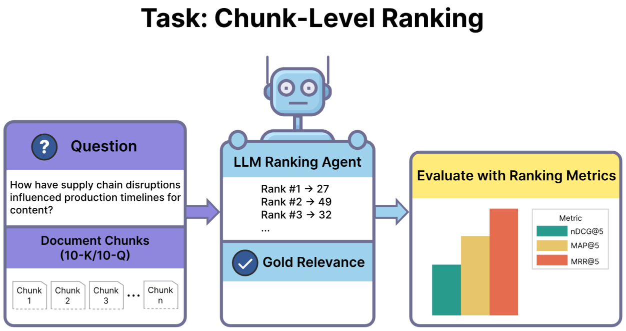

Task: Chunk-Level Ranking
?
Question
How have supply chain disruptions influenced production timelines for content?
Document Chunks (10-K/10-Q)
Chunk 1
Chunk 2
Chunk 3
...
Chunk n
LLM Ranking Agent
Rank #1 → 27
Rank #2 → 49
Rank #3 → 32
...
Gold Relevance
Evaluate with Ranking Metrics
Metric
nDCG@5
MAP@5
MRR@5

Figure 1: Task definition: given a financial QA question and all text chunks from a single filing, the goal is to rank chunks such that top-k evidence aligns with the question. The example illustrates input (question + chunks), agent's ranking output, and gold relevance labels used for evaluation.

A common agent-style method is to dump large batches of chunks into a large language model (LLM) and ask it to rank or select passages. In our setting this breaks down for two practical reasons. First, scale: multi-hundred-page reports exceed context budgets, and expanding the context window induces token and latency blow-ups that are costly and brittle. Second, opacity: monolithic, prompt-sensitive rankings are hard to audit because users

[*] Haipeng Zhang is the corresponding author.

AI4F '25, November 15, 2025, Singapore

cannot see why a passage was preferred, nor trace decisions back to concrete evidence, which is unacceptable in regulated financial analysis. Even strong zero-shot rerankers struggle to stay numerically and temporally grounded at paragraph level without domain structure [Choi et al. 2025a; Ma et al. 2023; Sun et al. 2023].

We argue for a different starting point: treat intra-document retrieval as a transparent, human-aligned review process rather than a single opaque prompt. Practitioners do not read an entire filing in one breath. A junior analyst first screens and bookmarks likely evidence; a senior analyst imposes a coherent global order; a committee resolves close calls under explicit criteria. We turn this workflow into a machine-usable contract by (i) recording each chunk as a structured *card* with entities, metrics, periods, numbers, section metadata, and verbatim spans and (ii) turning each question into a structured intent that states which entities/metrics/periods and whether numeric evidence are required. A lightweight, tournament-style review then mirrors the human process: a *junior* screen narrows candidates, a *senior* listwise review orders them globally, and a *committee* adjudicates near-ties; simple fusion and alignment rules translate these judgments into the final Top-*k*.

This human-aligned stance is not only a storytelling device; it constrains behavior in ways that matter for finance. (i) Numerical grounding improves because candidates without exact values are de-emphasized when the intent demands numbers. (ii) Temporal fit improves because fiscal expressions in intents must match normalized periods on cards. (iii) Boilerplate suppression follows from criteria that require entity/metric/period agreement. (iv) Auditability comes from card–intent traces: every selection is justified by explicit fields and verbatim spans rather than latent similarity. (v) Finally, cost predictability is achieved by staging small, purpose-built prompts instead of ever-growing monolithic contexts.

Our focus is the *intra-document* ranking setting: given a single, pre-selected filing, surface the most relevant chunks for a question. The example is in figure 1. This isolates the core retrieval bottleneck observed in financial QA, locating grounded evidence within long, heterogeneous documents before any generation step [Chen et al. 2021; Zhu et al. 2021]. While our method is model-agnostic and zero-shot (no task-specific fine-tuning or RL), it benefits from strict I/O contracts (deterministic decoding, JSON validation) that stabilize behavior and make runs reproducible.

By aligning the retrieval pipeline with how human analysts actually screen, order, and justify evidence, we can deliver early-precision improvements and desirable non-functional properties interpretability, auditability, and cost-efficiency without task-specific training.

We cast intra-document passage ranking as a transparent review process, with structured *cards* (supply of evidence) and *intents* (demand specification) as the primitives for decision making; **Lightweight tournament reranking.** we instantiate a junior–senior–committee workflow—batch screening, listwise ordering, pairwise adjudication—augmented by simple fusion and finance-aware alignment rules, yielding stable Top-*k* without long contexts; **Properties that matter in finance.** the pipeline exposes traceable rationales (card–intent matches and verbatim spans), reduces numeric/temporal mismatches, and keeps token/latency costs predictable in zero-shot settings; **Empirical validation.** on an intra-document ranking benchmark for financial QA, the human-aligned design consistently improves early retrieval quality over strong zero-shot baselines, while providing auditable traces and lower computational overhead.[1]

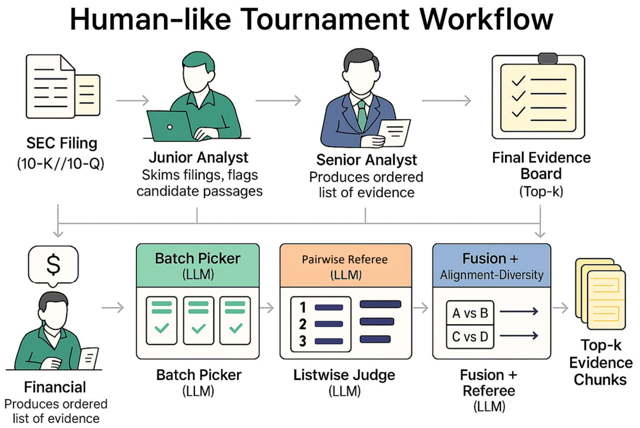


Figure 2: Comparison between a human financial analyst workflow (top) and the proposed LLM-based tournament pipeline (bottom). Junior analysts correspond to the Batch Picker, senior analysts to the Listwise Judge, and committee review to the Pairwise Referee, followed by fusion, alignment, and diversity adjustments. The diagram highlights the human-like design of the system.

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. 2025b] stressed retrieval-augmented generation in finance with expert triplets that reflect terse practitioner queries, and FinAgentBench [Choi et al. 2025a] 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 specifically targets this intra-document chunk-ranking setting and closes much of the gap in a zero-shot regime.

Retrieval and LLM-based reranking. Classical IR relies on lexical methods such as BM25, while modern pipelines incorporate dense retrievers and cross-encoders [Xiong et al. 2021]. Recently, LLMs

[1] Code and artifacts: https://github.com/XanderZhou2022/Human_Aligned_Multi_Agent

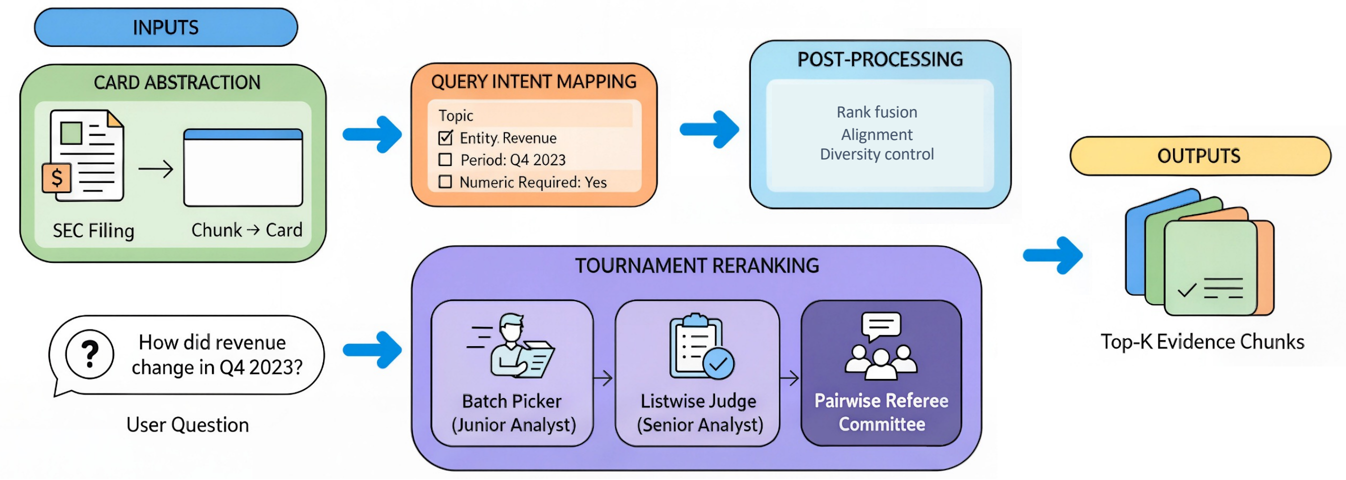


Figure 3: Pipeline overview: from SEC filing and question to card abstraction, query intent mapping, multi-agent tournament reranking, and Top-k evidence chunks.

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 [Ma et al. 2023; Sun 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. 2023], while setwise/tournament strategies mitigate order sensitivity and scale better with long lists [Chen et al. 2024; Zhuang et al. 2023]. Simple rank fusion such as reciprocal rank fusion (RRF) remains a strong baseline to aggregate noisy rankings [Cormack et al. 2009]. The present pipeline combines these insights with domain structure: batch-wise preselection and a listwise judge are followed by a pairwise referee, and outputs are fused with RRF, all guided by financial priors and numeric/temporal alignment signals.

Agentic and human-aligned reasoning. A parallel thread models how humans search and reason. ReAct interleaves reasoning and actions [Yao et al. 2022], Tree-of-Thoughts explores multiple solution paths [Yao et al. 2023], Reflexion adds self-critique with episodic memory [Shinn et al. 2023], and self-consistency improves chain-of-thought reliability [Wang et al. 2022]. 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 conclude that retrieval quality—not only model capacity—is the main failure point and that domain-aligned retrieval is crucial [Gao et al. 2023]. The present work fits this landscape by injecting structured cards and query intents into an agentic reranking process that mirrors analyst workflows (screening, global synthesis, and adjudication), improving grounding without any model updates.

3 METHOD

The task studied in this work is *intra-document passage ranking* for financial question answering, where the objective is to identify the most relevant chunks within a single long-form SEC filing (e.g., 10-K, 10-Q) that answer a given user query. This setting differs fundamentally from traditional cross-document retrieval in open-domain QA [Guu et al. 2020; Robertson and Zaragoza 2009]: instead of selecting among many short documents, the system must navigate within the boundaries of a single lengthy and heterogeneous report. Relevant evidence may be scattered across multiple sections, obscured by repeated boilerplate text, or embedded in dense numerical disclosures, making naïve keyword-based approaches insufficient.

The methodological challenge, therefore, is to balance fine-grained semantic matching with robustness to redundancy, temporal misalignment, and the idiosyncratic structure of financial filings. Existing retrieval methods such as BM25 [Robertson and Zaragoza 2009] or dense retrievers [Karpukhin et al. 2020] often either underperform on numerical reasoning or fail to account for domain-specific structure. In contrast, our design explicitly incorporates financial priors—entities, metrics, and periods—and organizes the retrieval process into a structured, multi-agent pipeline.

At the core of the approach, each filing is decomposed into compact *cards* that abstract key information such as entities, metrics, numerical values, and section metadata. Each question is mapped into a structured intent representation that specifies its topical focus, expected entities, metrics, and temporal scope. The ranking process is then staged as a sequence of agent-like modules: a *Batch Picker* acts as a junior analyst to filter candidates, a *Listwise Judge* provides a global ordering akin to a senior analyst's review, and a *Pairwise Referee* functions as a committee to adjudicate fine-grained comparisons. The outputs are further stabilized through reciprocal rank fusion, alignment scoring, and diversity controls, which integrate these perspectives into a final ranking.

This design is motivated by common failure modes in financial QA. Keyword-driven retrieval often elevates boilerplate passages with superficial matches while overlooking segments containing the required numerical evidence. Likewise, temporal ambiguity in financial reporting (e.g., interleaving of current and prior fiscal periods) frequently leads systems to surface outdated figures. By structuring the pipeline into interpretable stages and explicitly encoding numeric and temporal priors, the method aims to emulate how human analysts prioritize evidence: first narrowing the search space, then weighing candidates holistically, and finally resolving close calls through targeted comparisons. This multi-agent, tournament-inspired perspective builds on recent insights from LLM-based multi-agent reasoning [Du et al. 2023; Liang et al. 2024], adapting them to the challenges of financial disclosure analysis.

The remainder of this section details each component of the pipeline in turn.

3.1 Card Abstraction

The *card abstraction* module constitutes the foundation for zero-shot ranking over long financial filings. Its primary function is to transform raw text chunks from SEC reports (10-K and 10-Q) into compact, machine-readable artifacts—referred to as *cards*—that preserve essential semantic and numeric information while suppressing noise and boilerplate. Each card is instantiated as a structured JSON object with fields such as topic, entities, metrics, numbers, fiscal period, summary, and section metadata. In addition, the schema incorporates domain-level triples and verbatim evidence spans, ensuring that downstream ranking agents operate on normalized and auditable evidence rather than uncontrolled natural language.

The motivation for this design arises from the unique challenges of financial QA. Retrieval in this domain is particularly vulnerable to numeric drift, where approximate or reformatted values degrade precision [Chen et al. 2021]; temporal misalignment, where disclosures reference prior fiscal periods that do not correspond to the query; and boilerplate repetition, where legally mandated disclosures dominate lexical signals without providing substantive relevance. By enforcing exact numeric copying, explicit fiscal normalization, and boilerplate detection, the card abstraction directly mitigates these recurrent failure modes. Conceptually, the process parallels the work of a junior analyst preparing briefing notes: carefully recording numbers, dates, and contextual cues so that senior reviewers can later determine relevance.

Cards are constructed through a hybrid LLM–heuristic pipeline. An instruction-tuned model is prompted in a one-shot setting to produce strictly JSON-formatted outputs. The schema requires enumerated topical and section labels, verbatim evidence spans, and concise summaries capped at forty words. To ensure robustness, the pipeline enforces strict JSON-only input–output constraints, applies retry logic with exponential backoff, sanitizes keys to accommodate schema drift, and falls back to conservative placeholders when extraction fails. Each chunk therefore yields either a validated card or a placeholder aligned with the original index. This approach aligns with emerging practices in robust large language model extraction, where reliability is achieved through constrained decoding and schema validation [Ye et al. 2023].

To further enrich the cards, heuristic modules augment the model outputs. Named entity recognition identifies organizations, geographies, and executive roles [Devlin et al. 2019], while metric detectors highlight financial terms such as revenue, margin, or debt. Numeric sanitation distinguishes finance-specific tokens such as percentages and basis points from incidental values, and temporal heuristics normalize fiscal expressions into canonical forms (e.g., `FY2023-Q1`, `2022-12-31`). Boilerplate detection identifies repetitive sections such as Safe Harbor statements, allowing subsequent ranking modules to down-weight rather than discard them. This hybridization mimics the workflow of a junior analyst double-checking extracted notes against known patterns before passing them on to supervisors.

Each card undergoes schema validation to ensure consistency and quality. Constraints include non-empty entity and metric fields, fiscal period formats that match canonical calendar or fiscal expressions, length-limited summaries, and type checks for arrays and triples. Validation reports aggregate pass rates and issue categories, while per-chunk CSV files flag problematic cases for audit. This stage is analogous to compliance review in financial analysis, where briefing notes are checked for omissions or inconsistencies before being forwarded to decision makers.

The process outputs a `cards.jsonl` file containing one card per chunk, together with validation artifacts for quality control. These cards then serve as the evidence corpus for subsequent ranking agents: the Batch Picker (junior selection), the Listwise Judge (senior analyst), and the Pairwise Referee (committee). Formally, the abstraction can be expressed as a mapping

$$
\begin{aligned}
f : c_i &\mapsto \mathrm{Card}(c_i) \\
&= \big(T_i, E_i, M_i, N_i, P_i, S_i, D_i, \Xi_i\big).
\end{aligned}
\tag{1}
$$

where c_i is a text chunk, T_i denotes the topic, E_i the entity set, M_i the metrics, N_i the numeric values, P_i the normalized period, S_i the section label, D_i the domain triples, and Ξ_i the evidence spans. By transforming verbose filings into structured representations, the card abstraction enhances the feature richness available to tournament-style reranking while reducing susceptibility to common retrieval errors, thereby improving relevance, fidelity, and reproducibility in zero-shot financial question answering.

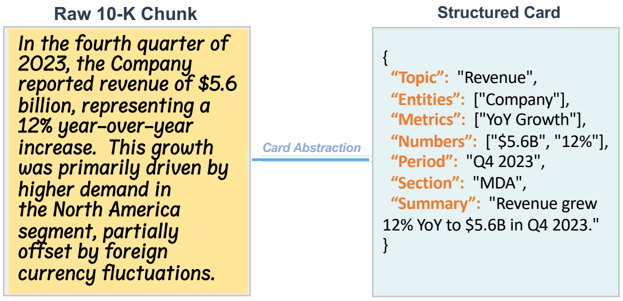

Raw 10-K Chunk
In the fourth quarter of 2023, the Company reported revenue of $5.6 billion, representing a 12% year-over-year increase. This growth was primarily driven by higher demand in the North America segment, partially offset by foreign currency fluctuations.
Card Abstraction
Structured Card
{
"Topic": "Revenue",
"Entities": ["Company"],
"Metrics": ["YoY Growth"],
"Numbers": ["$5.6B", "12%"],
"Period": "Q4 2023",
"Section": "MDA",
"Summary": "Revenue grew 12% YoY to $5.6B in Q4 2023."
}

Figure 4: Card abstraction: a raw 10-K chunk is transformed into a compact, structured card (topic, entities, metrics, numbers, period, section, summary).

3.2 Query Intent Mapping

On the demand side of the retrieval pipeline, user questions are converted into structured intent objects that expose the latent dimensions necessary for alignment with cards. This process, referred to as *query intent mapping*, transforms natural-language questions into compact representations that capture entities, metrics, temporal cues, numeric requirements, relational verbs, and lexical keywords. The result is a structured interface between heterogeneous questions and normalized card features, enabling downstream rankers to operate on aligned signals rather than raw text.

The motivation for query intent mapping stems from the inherent ambiguity of financial questions. For example, a query such as "How did revenue change last quarter?" implicitly demands numeric evidence (exact figures), a temporal anchor (the most recent fiscal quarter), and a relational operation (comparison over time). Without structured interpretation, retrieval systems risk surfacing irrelevant passages such as revenue recognition policies rather than numeric outcomes, or outdated references from prior fiscal years. Query intent mapping addresses this by decomposing questions into explicit dimensions that can be directly matched against card fields. In analogy, this module functions like a senior analyst clarifying a client's request: identifying precisely what evidence is required and producing a structured checklist that downstream agents can follow.

The mapping process is designed to be deterministic, ensuring reproducibility and cost efficiency without additional LLM inference. Intent objects are derived through lightweight analyzers that operate in parallel. Topic cues are extracted to map queries onto high-level financial themes such as liquidity, ESG, or revenue, while a controlled vocabulary of performance measures (e.g., EPS, margins, free cash flow) ensures normalization across variant phrasings. Entities such as organizations, executives, or geographies are detected using regex-based recognizers, while verbs are categorized into relational classes including increase, decrease, comparison, or explanation. Temporal anchors are normalized into canonical forms such as fiscal quarters or year-over-year expressions [Chen et al. 2021], and a numeric flag identifies whether the query requires explicit numerical evidence, with sensitivity to comparative contexts (YoY, QoQ). Finally, a capped set of lexical keywords is retained to support shallow lexical recall. Together, these features allow each question to be systematically converted into a structured intent record.

Formally, the mapping function $g(\cdot)$ transforms a question q_j into

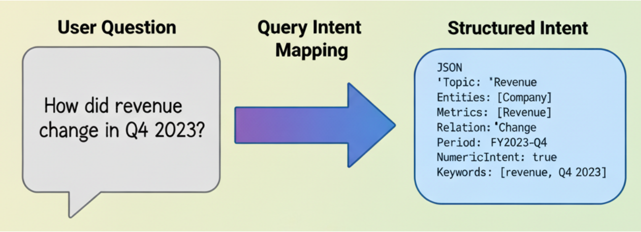
$$\text{Intent}(q_j) = g(q_j) = \left[T_j,\ E_j,\ M_j,\ R_j,\ \Theta_j,\ v_j,\ K_j\right]. \qquad (2)$$

where T_j denotes topical labels, E_j the entity set, M_j the metrics, R_j the relational class, Θ_j the temporal filters, v_j the numeric intent flag, and K_j the lexical keywords. By providing this explicit structure, the system ensures that demand-side expectations are clearly encoded for alignment with card features.

In addition to feature extraction, the module incorporates disambiguation and deduplication to ensure efficiency. Semantically equivalent queries are collapsed into a single canonical form before intent extraction, so that redundant or overlapping requests are consolidated. This mirrors the practice of a senior analyst who consolidates multiple client inquiries into a single clarified task before assigning it to the research team.

The benefits of query intent mapping extend to several recurrent failure modes in retrieval. Numeric grounding is enforced by linking queries that demand numbers to cards containing validated numeric evidence. Temporal misalignment is reduced by aligning fiscal expressions in the query with normalized period labels in the cards. Boilerplate contamination is avoided because repetitive disclosures generally lack the entity–metric overlap required by structured intent objects. Finally, query ambiguity is mitigated by decomposing vague natural-language prompts into explicit structured features, enabling downstream rankers to make more discriminative judgments.

The intent object therefore plays a complementary role to card abstraction. While cards provide structured evidence from financial filings, intent objects encode structured demand from natural-language questions. Their intersection forms the feature space consumed by the multi-agent ranking ensemble—comprising the Batch Picker, Listwise Judge, and Pairwise Referee. In this sense, query intent mapping serves as the interpretive bridge between heterogeneous user queries and structured financial evidence, ensuring that downstream ranking decisions are grounded in both the supply of reliable disclosures and the demand of precise question specifications.



```
JSON
'Topic: 'Revenue
Entities: [Company]
Metrics: [Revenue]
Relation:'Change
Period: FY2023-Q4
NumericIntent: true
Keywords: [revenue, Q4 2023]
```

Figure 5: Query intent mapping: a natural-language question is converted into a structured intent object (topic, entities, metrics, relation, period, numeric intent, keywords).

3.3 Zero-shot Tournament Reranking

This subsection details the end-to-end, zero-shot intra-document ranking procedure that selects the top-k evidence chunks from a single SEC filing for a given financial question. The pipeline assumes (i) a *card corpus* derived from filing chunks (Sec. 3.1) and (ii) a *structured query intent* extracted from the question (Sec. 3.2). Ranking proceeds through a three-stage multi-agent tournament—*Batch Picker* (junior analyst), *Listwise Judge* (senior analyst), and *Pairwise Referee* (committee)—followed by rank fusion, post-hoc alignment, and diversity control. The design mirrors a human review workflow: a junior analyst rapidly narrows the search space, a senior analyst produces a coherent global order, and a committee resolves close calls through targeted comparisons. The inspiration for such multi-agent pipelines is consistent with prior work on ensemble and committee-based ranking [Croft et al. 2010; Voorhees and Tice 1999].

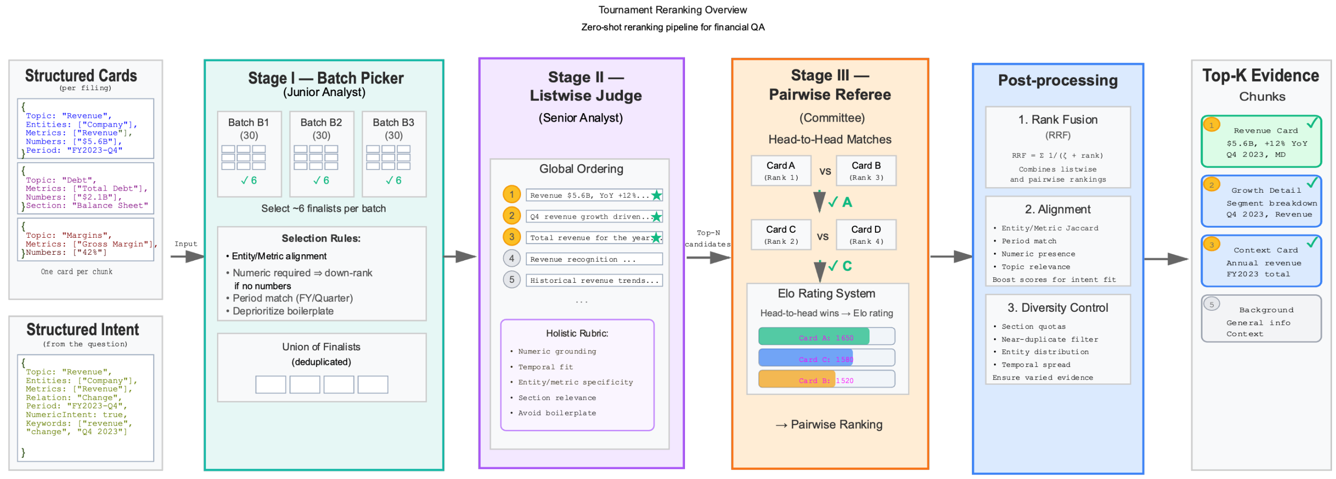

Tournament Reranking Overview
Zero-shot reranking pipeline for financial QA
Structured Cards
(per filing)
{ Topic: "Revenue", Entities: ["Company"], Metrics: ["Revenue"], Numbers: ["$5.6B"], Period: "FY2023-Q4" }
{ Topic: "Debt", Metrics: ["Total Debt"], Numbers: ["$2.1B"], Section: "Balance Sheet" }
{ Topic: "Margins", Metrics: ["Gross Margin"], Numbers: ["42%"] }
One card per chunk
Structured Intent
(from the question)
{ Topic: "Revenue", Entities: ["Company"], Metrics: ["Revenue"], Relation: "Change", Period: "FY2023-Q4", NumericIntent: true, Keywords: ["revenue", "change", "Q4 2023"] }
Input
Stage I — Batch Picker
(Junior Analyst)
Batch B1 (30) ✓6
Batch B2 (30) ✓6
Batch B3 (30) ✓6
Select ~6 finalists per batch
Selection Rules:
• Entity/Metric alignment
• Numeric required ⇒ down-rank if no numbers
• Period match (FY/Quarter)
• Deprioritize boilerplate
Union of Finalists (deduplicated)
Stage II — Listwise Judge
(Senior Analyst)
Global Ordering
1 Revenue $5.6B, YoY +12%...
2 Q4 revenue growth driven...
3 Total revenue for the year...
4 Revenue recognition ...
5 Historical revenue trends...
...
Holistic Rubric:
• Numeric grounding
• Temporal fit
• Entity/metric specificity
• Section relevance
• Avoid boilerplate
Top-N candidates
Stage III — Pairwise Referee
(Committee)
Head-to-Head Matches
Card A (Rank 1) vs Card B (Rank 3)
✓ A
Card C (Rank 2) vs Card D (Rank 4)
✓ C
Elo Rating System
Head-to-head wins → Elo rating
Card A: 1650
Card C: 1580
Card B: 1520
→ Pairwise Ranking
Post-processing
1. Rank Fusion (RRF)
RRF = Σ 1/(ζ + rank)
Combines listwise and pairwise rankings
2. Alignment
• Entity/Metric Jaccard
• Period match
• Numeric presence
• Topic relevance
Boost scores for intent fit
3. Diversity Control
• Section quotas
• Near-duplicate filter
• Entity distribution
• Temporal spread
Ensure varied evidence
Top-K Evidence
Chunks
1 Revenue Card $5.6B, +12% YoY Q4 2023, MD ✓
2 Growth Detail Segment breakdown Q4 2023, Revenue ✓
3 Context Card Annual revenue FY2023 total ✓
5 Background General info Context

Figure 6: Analogy between a human financial analyst workflow (top) and the proposed LLM-based tournament pipeline (bottom). Junior analysts correspond to the Batch Picker, senior analysts to the Listwise Judge, and committee review to the Pairwise Referee, followed by post-hoc adjustments. This illustrates the human-like design of the system.

3.3.1 *Preliminaries.*

Let $C = \{c_1, \ldots, c_n\}$ denote the set of cards for the selected filing. Each card c contains structured fields

$$c = \big(\text{topic}, \text{section}, \text{summary}, E_c, M_c, N_c, P_c, \text{boilerplate}\big). \tag{3}$$

with entities E_c, metrics M_c, numbers N_c, and a normalized period P_c. Let the intent object for question q be

$$\text{Intent}(q) = (T_q, E_q, M_q, R_q, \Theta_q, \nu_q, K_q), \tag{4}$$

where E_q, M_q denote demanded entities and metrics, Θ_q are period cues (e.g., fiscal tags), $\nu_q \in \{0, 1\}$ indicates whether numeric evidence is required, and K_q are lexical keywords. The task is to output an ordered list π over the chunk indices associated with C, truncated to the top-k.

The design explicitly addresses recurrent challenges in financial QA. *Numeric drift* is curtailed by prioritizing cards with exact values when $\nu_q = 1$. *Temporal misalignment* is mitigated via a compatibility term that penalizes periods inconsistent with Θ_q [Chen et al. 2021]. *Boilerplate redundancy* is reduced by down-weighting cards flagged as boilerplate and enforcing section-level diversity. These measures ensure that retrieved chunks resemble the evidence board of a human analyst: numerically faithful, time-appropriate, and non-redundant.

3.3.2 *Stage I: Batch Picker (Junior Analyst).*

Cards are first partitioned into disjoint groups $\mathcal{B}_1, \ldots, \mathcal{B}_G$. For each group $\mathcal{B}_g$, a deterministic LLM prompt instructs a "junior analyst" to select a small set of promising *finalists* $\mathcal{F}_g \subset \mathcal{B}_g$. The rubric emphasizes (i) entity/metric overlap with (E_q, M_q), (ii) numeric presence if $\nu_q = 1$, (iii) period compatibility with Θ_q, and (iv) deprioritization of boilerplate. To reduce variance, groups may be reshuffled and evaluated across multiple rounds, with the union of picks retained:

$$\mathcal{F} = \bigcup_{g=1}^{G} \mathcal{F}_g, \text{(deduplicated)}. \tag{5}$$

This stage mirrors a junior analyst skimming sections and flagging likely evidence snippets, ensuring wide coverage before deeper review.

3.3.3 *Stage II: Listwise Judge (Senior Analyst).*

A "senior analyst" LLM then receives the finalist set $\mathcal{F}$ and returns a *global order* π_{L} over a capped subset of $\mathcal{F}$ to preserve context fidelity. Instructions emphasize holistic coherence: aligning entities and metrics, preferring exact numerics when required, enforcing temporal consistency, and avoiding boilerplate-driven matches. This listwise step corrects for the local myopia of Stage I and provides a coherent global ordering, akin to a senior analyst's structured review.

3.3.4 *Stage III: Pairwise Referee (Committee).*

Borderline cases among the top of π_{L} are adjudicated through targeted pairwise comparisons. A scheduler produces a sparse match list $\mathcal{M} \subset \mathcal{F} \times \mathcal{F}$, bounding the number of comparisons per item. For each match $(i, j) \in \mathcal{M}$, a "committee" LLM returns a winner $w \in \{i, j\}$ under the same rubric but with head-to-head focus. Item ratings are updated via Elo-style dynamics [Elo 1978]:

$$\begin{aligned} \hat{p}_i &= \sigma\Big(\frac{r_i - r_j}{\tau}\Big), \\ r_i &\leftarrow r_i + \kappa\,(s_i - \hat{p}_i), \\ r_j &\leftarrow r_j + \kappa\,\big((1 - s_i) - (1 - \hat{p}_i)\big). \end{aligned} \tag{6}$$

where r_i is the rating for item i, $s_i \in \{0, 1\}$ is the match outcome, σ is the logistic function, τ a temperature, and κ a step size. Sorting items by final r_i yields the pairwise order π_{P}. This step parallels a committee of experts resolving close calls by direct comparison.

3.3.5 *Rank Fusion, Alignment, and Diversity.*

The outputs of listwise and pairwise ranking are combined through Reciprocal Rank Fusion (RRF) [Voorhees and Tice 1999]:

$$\text{RRF}(c) = \sum_{o \in \{\pi_{\text{L}}, \pi_{\text{P}}\}} \frac{1}{\zeta + \text{rank}_o(c)}. \tag{7}$$

with ζ moderating tail effects. Post-hoc alignment further adjusts scores by incorporating domain-specific priors:

$$\begin{aligned}\mathrm{Align}(c,q) = {} & \alpha_E \cdot \mathrm{Jacc}(E_c, E_q) \\ & + \alpha_M \cdot \mathrm{Jacc}(M_c, M_q) \\ & + \alpha_P \cdot \mathbf{1}\{\mathrm{PeriodMatch}(P_c, \Theta_q)\} \\ & + \alpha_N \cdot v_q \cdot \mathbf{1}\{|N_c| > 0\}.\end{aligned} \tag{8}$$

The final score is

$$S(c \mid q) = \mathrm{RRF}(c) + \lambda \cdot \mathrm{Align}(c,q). \tag{9}$$

where alignment rewards entity, metric, temporal, and numeric fidelity.

Diversity constraints prevent redundancy by imposing per-section quotas and suppressing near-duplicate neighbors within a section. The final output is a compact, non-redundant top-k set of chunks that balances precision with coverage.

Algorithm 1 Zero-shot Tournament Reranking within a Filing

```
Require: Card set $C$ for a filing, query $q$, target cutoff $k$
Partition $C$ into batches $\{\mathcal{B}_1, \ldots, \mathcal{B}_G\}$
$\mathcal{F} \leftarrow \emptyset$
for $g \leftarrow 1$ to $G$ do
    $\mathcal{F}_g \leftarrow$ LLM_PickTop($q$, $\mathcal{B}_g$)   ▷ Junior analyst picks finalists
    $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_g$
end for
$\mathcal{F} \leftarrow$ Deduplicate($\mathcal{F}$)
$\pi^{\mathrm{L}} \leftarrow$ LLM_ListwiseRank($\mathcal{F}, q$)   ▷ Senior analyst global order
$\mathcal{H} \leftarrow$ Top($\pi^{\mathrm{L}}, h$)   ▷ Head-to-head pool
$\mathcal{M} \leftarrow$ SchedulePairs($\mathcal{H}$)
for all $c_i \in \mathcal{H}$ do
    $r_i \leftarrow r_0$
end for
for all $(i, j) \in \mathcal{M}$ do
    $w \leftarrow$ LLM_PairwiseWinner($q$, $c_i$, $c_j$)
    $(r_i, r_j) \leftarrow$ Elo_Update($r_i$, $r_j$, $w$)
end for
$\pi^{\mathrm{P}} \leftarrow$ SortByScore($\{(c_i, r_i)\}$)
for all $c \in \mathcal{F}$ do
    score[$c$] $\leftarrow \mathrm{RRF}(c) + \lambda \cdot \mathrm{Align}(c, q)$
end for
$\pi \leftarrow$ StableMerge($\pi^{\mathrm{L}}, \pi^{\mathrm{P}}$, score)
$\pi \leftarrow$ DiversifyAndDedup($\pi$)
return Top($\pi$, $k$)
```

3.3.6 Complexity and Reliability Considerations. Let $n = |C|$, G the number of batches, and $|\mathcal{H}|$ the head-to-head pool size. The picker requires G LLM calls per round, the listwise judge one call, and the referee $O(|\mathcal{M}|)$ calls with $|\mathcal{M}|$ proportional to $|\mathcal{H}|$. All LLM interactions enforce strict JSON schema validation, retries, deterministic decoding, and fallbacks to preserve reproducibility. These safeguards are essential in financial QA, where interpretability and auditability are as critical as accuracy.

4 EXPERIMENTS

4.1 Experimental Setup

We evaluate the proposed zero-shot, intra-document chunk-ranking pipeline on the **FinAgentBench** benchmark [Choi et al. 2025a], which consists of financial QA tasks derived from U.S. SEC filings (10-K and 10-Q reports). For each query, the system is provided with a single pre-selected filing and must rank its constituent chunks, surfacing the top-k passages most likely to answer the question. This setting isolates the challenge of *intra-document* retrieval—locating relevant financial evidence within long and heterogeneous reports, where pertinent information may be interleaved with boilerplate or stale disclosures.

All experiments are conducted in a Google Colab environment with Python 3.x, using the OpenAI Chat Completions API as the LLM backbone. Unless otherwise noted, all runs fix random seeds to 42, decoding temperature to 0, and enforce strict JSON parsing for robustness. We report average results across benchmark queries; bootstrap confidence intervals were also computed but are omitted for brevity.

4.2 System Variants

We compare three progressively refined variants of the pipeline, corresponding to development stages.

V1: Baseline (Prompted Listwise Reranker). The baseline constructs compact *cards* for each chunk, consisting of normalized fields such as entities, metrics, numbers, and summaries. A single-stage listwise reranker LLM then orders all candidate cards directly using handcrafted prompt rules. This version establishes the benefit of card abstraction and zero-shot prompting but lacks mechanisms for stabilizing rankings or suppressing redundancy.

V2: Prompt Optimization with Batch Picker. The second variant introduces a two-stage tournament. First, a *Batch Picker* evaluates disjoint groups of cards (batch size $B = 30$), selecting $m = 6$ finalists per group. The union of finalists is then passed to a *Listwise Judge* for global ordering. Prompts are further optimized: explicit instructions down-rank boilerplate, prioritize numeric evidence when the question implies quantitative intent, and enforce temporal filters (e.g., fiscal quarters). These refinements address recurrent financial QA failure modes [Chen et al. 2021], improving alignment between questions and evidence.

V3: Pairwise Referee with Fusion and Alignment. The final variant adds a *Pairwise Referee*, inspired by tournament decision-making and prior applications of head-to-head comparison in ranking [Elo 1978; Joachims 2002]. The top-20 candidates from the listwise stage are compared in scheduled pairwise matches, with outcomes aggregated via Elo-style rating. Rankings from the listwise and pairwise stages are fused using Reciprocal Rank Fusion (RRF) [Voorhees and Tice 1999]. Post-hoc adjustments further improve alignment by rewarding entity/metric/period matches, while diversity constraints enforce section quotas and suppress near-duplicate evidence. This version represents the full multi-agent pipeline, designed to maximize early precision while maintaining topical coverage.

4.3 Evaluation Metrics

We adopt three widely used information retrieval metrics [Baeza-Yates and Ribeiro-Neto 1999; Järvelin and Kekäläinen 2002]. Normalized Discounted Cumulative Gain at rank 5 (nDCG@5) measures graded relevance while emphasizing early ranks. Mean Average Precision at rank 5 (MAP@5) quantifies precision across the top-*k* results. Mean Reciprocal Rank at rank 5 (MRR@5) reflects how quickly the first relevant chunk appears. All metrics are reported as averages across evaluation queries.

4.4 Results

Table 1 reports the performance of our three system variants alongside the baseline results from FinAgentBench [Choi et al. 2025a]. The benchmark baselines include GPT-o3, Claude-Opus-4, and Claude-Sonnet-4, as evaluated by Wang et al. (2025) on the same chunk-ranking task. Our results are not directly comparable in training regime, since we use a strictly zero-shot setting without reinforcement fine-tuning, but the comparison provides context for the effectiveness of the proposed pipeline.

Table 1: Performance on intra-document chunk ranking. FinAgentBench baselines are reported from [Choi et al. 2025a]. Our variants are evaluated in a zero-shot setting. Best scores within each block are highlighted.

Model	nDCG@5	MAP@5	MRR@5
FinAgentBench Baselines			
GPT-o3	0.351	0.257	0.538
Claude-Opus-4	0.418	**0.307**	**0.568**
Claude-Sonnet-4	**0.419**	0.296	0.567
Our Zero-shot Pipeline Variants			
V1: Baseline (Prompt)	0.4225	0.6892	0.7107
V2: Prompt Optimization	0.4381	0.6979	0.7334
V3: Pairwise Referee	**0.4618**	**0.7114**	**0.7659**

4.5 Analysis

The comparison reveals several trends. First, our baseline variant (V1) already outperforms the strongest FinAgentBench baseline (Claude-Sonnet-4) in all three metrics, despite operating in a strictly zero-shot regime without task-specific fine-tuning. This highlights the importance of structured card abstraction and tailored prompt design for financial QA. Second, successive refinements further improve results: prompt optimization in V2 yields gains of +3.7% nDCG@5 and +3.2% MRR@5 relative to V1, while the introduction of the pairwise referee in V3 provides the largest overall improvements (+7.8% MRR@5 over V1). These improvements confirm that multi-agent tournament-style reranking is particularly effective at resolving near-ties and stabilizing rankings.

Interestingly, the MAP@5 values for our systems are considerably higher than those of the benchmark baselines. This is largely due to the structured pipeline's ability to suppress boilerplate and surface multiple relevant passages within the top-*k*. However, the more modest gains from V2 to V3 in MAP@5 suggest a trade-off: while pairwise adjudication sharpens early precision (nDCG, MRR), diversity constraints reduce the number of near-duplicate relevant chunks captured, which in turn caps precision. Overall, the results validate the effectiveness of the proposed design, showing that even without training, careful structuring and prompt engineering can surpass strong baseline models.

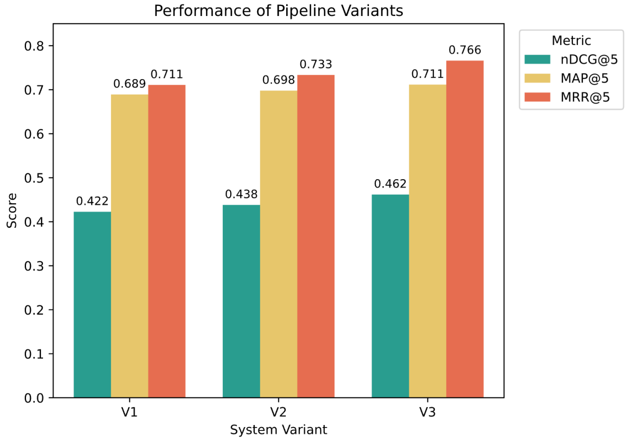


Figure 7: Performance of pipeline variants (V1–V3). Each refinement step improves ranking metrics (nDCG@5, MAP@5, MRR@5), illustrating the benefit of prompt optimization and pairwise reranking.

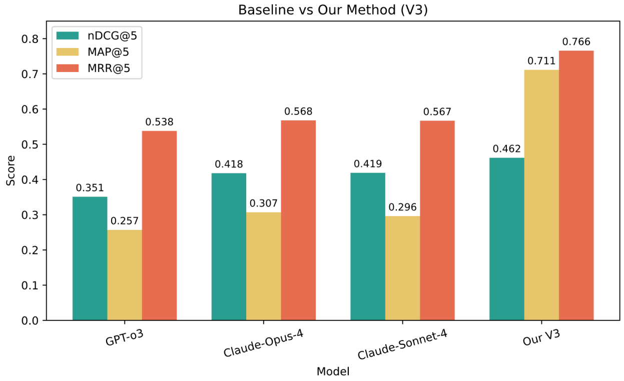


Figure 8: Comparison against strong baselines (GPT-o3, Claude-Opus-4, Claude-Sonnet-4). The proposed pipeline (V3) achieves higher precision under zero-shot conditions.

5 DISCUSSION AND FUTURE WORK

The proposed tournament-style intra-document reranking framework demonstrates that zero-shot large language models (LLMs) can be organized into multi-agent pipelines to overcome common retrieval challenges in long financial filings. By explicitly addressing numeric grounding, temporal fit, and boilerplate suppression, the approach outperforms both baseline language models and previously reported benchmarks such as GPT-o3 and Claude models on the FinAgentBench dataset [Choi et al. 2025a]. These results suggest that structured prompting and staged decision-making offer a promising alternative to costly fine-tuning or retrieval-augmented architectures.

Despite these advances, several limitations remain. First, the pipeline incurs non-trivial computational cost, particularly in the pairwise referee stage, which scales quadratically with the number of finalists. Although scheduling heuristics and reciprocal rank fusion mitigate this, efficiency remains a concern for large-scale deployment. Second, the current approach is restricted to single-document retrieval, whereas many real-world financial QA scenarios require reasoning across multiple filings or even heterogeneous sources such as press releases and earnings calls. Extending the framework to multi-document retrieval with effective evidence aggregation is an important direction for future work. Third, while our zero-shot design avoids task-specific training, it may still be sensitive to prompt formulations and schema drift; lightweight fine-tuning or reinforcement learning could further stabilize performance.

Another open avenue lies in the integration of retrieval-augmented generation (RAG). Recent work highlights the potential of coupling retrieval pipelines with generative reasoning to improve both precision and interpretability in financial tasks [Xu et al. 2023]. Embedding the proposed ranking module as a retrieval front-end to RAG architectures could enhance answer quality while retaining traceability. Moreover, adopting calibration techniques to quantify uncertainty in rankings [Kamath et al. 2020] may improve the robustness of decision support in high-stakes domains.

Finally, a qualitative case study illustrates how the pipeline corrects typical baseline failures. For instance, when asked about recent revenue guidance, generic LLMs often return sections on revenue recognition policies, whereas our system prioritizes MD&A outlook passages containing forward-looking statements and numeric projections. Such examples provide intuitive evidence for the pipeline's effectiveness and highlight its alignment with analyst-style reasoning.

In summary, while the proposed system achieves measurable gains over strong zero-shot baselines, it remains an early step toward fully robust financial QA. Future work will explore multi-document extensions, integration with generative reasoning, and adaptive fine-tuning strategies, paving the way for retrieval systems that are both accurate and practical in real-world financial analysis.

REFERENCES

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval.* Addison-Wesley, Boston, MA, USA.

Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. 2024. TourRank: Utilizing Large Language Models for Documents Ranking with a Tournament-Inspired Strategy. *arXiv preprint arXiv:2406.11678* (2024).

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.* Association for Computational Linguistics, Online, 3697–3711.

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.* Association for Computational Linguistics, 10458–10473.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025a. FinAgentBench: Agentic Retrieval for Financial Document Understanding. *arXiv preprint arXiv:2508.14403* (2025).

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy-yong Sohn, and Alejandro Lopez-Lira. 2025b. FinDER: Financial Dataset for Question Answering and Evaluating Retrieval-Augmented Generation. *arXiv preprint arXiv:2504.15800* (2025).

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.* ACM, 758–759.

W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search Engines: Information Retrieval in Practice.* Addison-Wesley.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL.* 4171–4186.

Yilun Du, Shuang Li, et al. 2023. Improving Factuality and Reasoning in Language Models through Multi-Agent Debate. In *Proceedings of NeurIPS.*

Arpad E. Elo. 1978. *The Rating of Chessplayers, Past and Present.* Arco Publishing.

Tianyu Gao et al. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2023).

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval-Augmented Language Model Pre-Training. In *Proceedings of ICML.* 3929–3938.

Yupan Huang et al. 2022. LayoutLMv3: Pre-Training for Document AI with Unified Text and Image Masking. In *Proceedings of ACL.*

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. FinanceBench: A New Benchmark for Financial Question Answering. arXiv:2311.11944 [cs.CL] https://arxiv.org/abs/2311.11944 arXiv preprint.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

Thorsten Joachims. 2002. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)* (Edmonton, Alberta, Canada). Association for Computing Machinery, New York, NY, USA, 133–142. https://doi.org/10.1145/775047.775067

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective Question Answering under Domain Shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).* Association for Computational Linguistics, Online, 5684–5696. https://doi.org/10.18653/v1/2020.acl-main.503

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, Miami, Florida, USA, 17889–17904. https://doi.org/10.18653/v1/2024.emnlp-main.992

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-Shot Listwise Document Reranking with a Large Language Model. *arXiv preprint arXiv:2305.02156* (2023).

Ahmed Masry et al. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Proceedings of ACL.*

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. 2023. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. *arXiv preprint arXiv:2306.17563* (2023).

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009), 333–389.

Noah Shinn et al. 2023. Reflexion: An Autonomous Agent with Dynamic Memory and Self-Reflection. *arXiv preprint arXiv:2303.11366* (2023).

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. *arXiv preprint arXiv:2304.09542* (2023).

Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 Question Answering Track Report. In *Proceedings of TREC.*

Xuezhi Wang et al. 2022. Self-Consistency Improves Chain-of-Thought Reasoning in Language Models. In *Proceedings of ICLR.*

Lee Xiong et al. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *Proceedings of ICLR.*

Yilun Xu et al. 2023. RetrievalBench: A Benchmark for Evaluating Retrieval in Large Language Models. In *Proceedings of EMNLP.*

Shunyu Yao et al. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint arXiv:2210.03629* (2022).

Shunyu Yao et al. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint arXiv:2305.10601* (2023).

Qinyuan Ye, Xiang Chen, and Xiang Ren. 2023. Large Language Models Are Zero-Shot Extractors. *Transactions of the Association for Computational Linguistics* 11 (2023), 690–706.

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 Data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 3277–3287.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2023. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. *arXiv preprint arXiv:2310.09497* (2023).

A PROMPTS AND TEMPLATES

This appendix lists the exact prompts used in our pipeline, including card abstraction, batch picking, listwise ranking, pairwise judging, and the lexical resources for intent mapping. All prompts enforce strict JSON-only I/O to ensure reproducibility.

A.1 Card Abstraction Prompt

```
CARD_SYS = "You␣are␣a␣meticulous␣financial␣analyst␣and␣
     information␣extractor.␣Output␣STRICT␣JSON␣only."

CARD_USER_TMPL = """From the chunk below, produce a
     compact CARD capturing WHAT the text talks about.

Strict rules:
1) Copy numbers EXACTLY as in text (no rounding).
2) Use ENUMS for topic/section only.
3) Evidence spans MUST be copied verbatim.
4) DO NOT invent facts or entities that are not in the
     text.
5) If the chunk mentions any proper nouns ... [truncated
     for space]

Return JSON EXACTLY as:
{
  "card": {
    "topic": "<ENUM>",
    "entities": ["..."],
    "metrics": ["..."],
    "numbers": [],
    "period": "",
    "section": "<ENUM>",
    "summary": "<= 40 words factual"
  },
  "domain": {...},
  "evidence_spans": {...}
}

Chunk (first 6000 chars):
<<CHUNK>>

Section_hint (optional): <<SECTION_HINT>>
"""
```

A.2 Batch Picker (Group Elimination) Prompt

```
sys = (
  "You␣are␣a␣concise␣financial␣retrieval␣judge.␣Output␣
      STRICT␣JSON␣only.␣"
  "Judge␣relevance␣for␣answering␣the␣question."
)

usr = f"""Question:
{question_text}

You will receive candidate chunks as compact 'cards' with
      fields:
[chunk_uid, topic, section, period, entities, metrics,
     numbers, summary, boilerplate flag]

Instructions:
- Rank the chunks by how well they answer the question.
- Respect temporal intent ...
- Return STRICT JSON exactly as: {"top": ["<chunk_uid>",
     ...]}
"""
```

A.3 Listwise Judge Prompt

```
sys = "You␣are␣a␣concise␣financial␣retrieval␣judge.␣
     Output␣STRICT␣JSON␣only."

usr = f"""Question:
{question_text}

FINALIST chunks (cards):
{json.dumps(finalists, ensure_ascii=False)}

Rank by: ability to answer the question, temporal fit,
     specificity, numeric evidence (when implied).
Return STRICT JSON: {"top_k": ["<chunk_uid>", ...]} (max
     {k} items).
"""
```

A.4 Pairwise Referee Prompt

```
sys = (
  "You␣are␣a␣concise␣financial␣retrieval␣judge.␣Output␣
      STRICT␣JSON␣only.␣"
  "Judge␣relevance␣for␣answering␣the␣question."
)

usr = f"""Question:
{question_text}

Candidates: {json.dumps(items, ensure_ascii=False)}

Instructions:
- Compare two chunks head-to-head.
- Respect numeric/temporal constraints.
- Return STRICT JSON exactly as: {"winner": "<chunk_uid
     >"}
"""
```

A.5 Lexical Resources for Intent Mapping

```
TOPIC_ENUM = {"Risk","ESG","Market","Revenue","
     Profitability","Liquidity",
              "Costs/Expenses","Guidance/Outlook","
                   AccountingPolicy","Legal",
              "MD&A","FinancialStatements","Other"}

METRIC_VOCAB = {
    "revenue","sales","eps","earnings","margin","opex","
         sg&a","r&d","cogs",
    "capex","cash_flow","fcf","inventory","backlog","
         bookings","units","asp",
    "debt","interest","liquidity","guidance","outlook","
         churn","retention",
    "headcount","hiring","layoffs","wage","salary","esg",
         "carbon","risk","legal"
}

RELATION_LEX = {
    "increase": {"increase","rise","grow","expand"},
    "decrease": {"decrease","decline","drop","contract"},
    "influence": {"impact","affect","drive","cause"},
    "compare": {"yoy","qoq","versus","compared"},
    "explain": {"why","reason","driver","attribution"}
}
```