金融 RAG 毕业论文（定稿草稿）

摘要

金融问答（FinDER）任务中的复杂查询常涉及多年份、多实体与显式数值计算。传统单步检索+占位式生成在复杂问题上易出现证据覆盖不足与算术错误。本文构建了一个可复现的金融 RAG 系统，在无外部 LLM API 的约束下，引入规则驱动的多步检索与显式计算器，并通过门控策略与系统化调参避免性能回退。实验结果表明：检索器微调显著提升整体检索表现（full dev Recall@10: 0.3246→0.3772）；多步检索在复杂子集上维持不退化且 MRR 略有提升；计算器通过门控确保数值指标不下降，为后续提升奠定稳定基线。

关键词：金融问答；检索增强生成；多步检索；显式计算；误差分析；可复现

---

1 引言

金融问答（FinDER）场景中的查询往往具有**高信息密度**与**强对比/计算需求**：同一问题可能同时涉及多个年份、实体、指标，并要求对证据进行对齐与推理。传统单步检索 + 占位式生成容易在复杂问题上出现**证据覆盖不足**与**算术错误**。

为此，本文围绕可复现的金融 RAG（Retrieval-Augmented Generation）系统，构建并验证了一个分层可控的工程方案：在强约束（无外部 LLM API）的条件下，引入**规则驱动的多步检索**与**显式计算器**，并通过系统化调参与门控机制避免性能回退。

本文贡献如下：

可复现工程框架：建立从数据规范化、检索、评测到实验编排的一体化体系，所有实验产出统一落盘，可审计、可回滚。

- **多步检索与门控策略**：实现 gap 检测、合并策略与停止规则，构建可控的多步检索循环，保障复杂查询的证据覆盖。

显式计算器：在证据抽取与单位/年份校验基础上进行程序化计算，并通过门控阈值降低算错风险。

- **系统化调参与对照分析**：输出 full dev / complex dev / numeric dev 的对照与消融结果，支持论文级结果表格与错误分析。

> 说明：请求中提到的 `my-thesis/baseline.pdf` 在当前仓库未找到，本文结构参考现有 Step6 结果与工程文档完成。

---

2 相关工作

本研究与以下方向密切相关：

金融问答与金融文本理解
金融 QA 数据集与任务通常聚焦于财报、公告、研报等长文档环境中的事实与数值问答。FinDER 等数据集强调对证据的精确定位与多字段对齐，对检索与推理能力要求较高。

检索增强生成（RAG）
RAG 框架通过检索模块提供高相关证据，再由生成模块构造答案。近期研究关注**密集检索、稀疏检索与混合检索**的结合，以及检索器微调对下游 QA 的传导效果。

多跳/多步检索与推理
多跳检索强调通过多轮检索逐步完善证据覆盖，常见于复杂比较问题与跨文档推理问题。本文的多步检索采用规则驱动的 gap 检测与可控门控策略，以保证解释性和稳定性。

显式数值推理与计算
数值类问题的核心挑战是单位、年份与数值的对齐。显式计算器通过结构化抽取与程序化计算降低算术错误，并配合门控策略避免错误传播。

---

3 方法

本文系统由四个核心模块组成：**查询理解**、**多步检索推理**、**证据整合与计算**、**答案生成**。整体流程如下：

```
flowchart TD
  Q[Query] --> U[Query Understanding]
  U --> R1[Step-1 Retrieval]
  R1 --> G[Gap Detector]
  G -->|gap & gate| Rn[Refined Query Retrieval]
  Rn --> M[Merge & Rank]
  M --> C[Calculator (optional)]
  C --> A[Template Answer]
  G -->|no gap| M
```

1. 查询理解
通过规则与词典对查询进行初步解析，包括年份识别、比较关系识别、数值提示词识别等。该模块用于驱动多步检索的 gap 类型判断与后续计算任务类型预测。

2. 多步检索推理
多步检索在每一步使用当前查询进行检索，随后基于 gap 检测决定是否继续检索。核心机制包括：

Gap Detector：检测年份缺失、实体缺失等信息缺口。
- **Gate（门控）**：当 gap_conf < min_gap_conf 时，停止后续检索（避免 query 漂移）。

Merge Strategy：采用 `maxscore` 或 `step1_first` 合并策略对跨步候选排序与截断。
- **Stop Criteria**：达到 max_steps 或无新增证据时停止。

Step6 最优多步配置（dev）：
max_steps=2
- top_k_each_step=10

novelty_threshold=0.0
- stop_no_new_steps=1

merge_strategy=maxscore
- gate.min_gap_conf=0.3

3. 证据整合与计算
该模块包含数值抽取与显式计算：

抽取：从证据文本中提取数值、单位、年份，并标注 inferred_year 与 confidence。
- **计算器**：支持 YoY / 差值 / 占比 / 倍数等任务。对于单位不一致、年份缺失、候选冲突等情况进行拒算，并记录原因。

Step6 最优门控（numeric dev）：
min_conf=0.2
- allow_task_types=[]（在当前版本中关闭计算任务以避免 Numeric-EM 回退）

4. 答案生成
生成采用模板化策略：
若计算器返回 `status=ok` 且通过门控，则输出结构化计算结果与简要解释；
- 否则回退到 baseline 的占位式生成，并记录 fallback 原因。

---

4 实验设置

数据集与划分
使用 FinDER 数据集，按官方或既有切分方式划分为 train / dev / test。所有子集与样本格式统一为：

```
{
  "qid": "...",
  "query": "...",
  "answer": "...",
  "evidences": [{"text": "...", "doc_id": null, "meta": {}}],
  "meta": {}
}
```

子集定义
complex_dev：满足任一条件即进入子集：
- - 多证据（evidence ≥ 2）
- - 查询包含 ≥2 年份

- - 查询含比较/变化关键词（vs/compare/yoy/增长率 等）
- - 查询含数值与年份组合
- **numeric_dev**：查询或答案含数值/百分号/同比/差值/倍数关键词。

评价指标
检索指标：Recall@k、MRR@k、evidence_hit@k
- **数值指标**：Numeric-EM、相对误差（RelErr）、覆盖率（Coverage）
不确定匹配比例：当 doc_id/evidence_id 缺失时，回退到文本匹配并记录比例。
-
- 关键参数
- 检索器：稀疏（BM25）+ 稠密（sentence-transformers）+ 混合（alpha=0.5）
多步检索（best）：max_steps=2, top_k_each_step=10, merge_strategy=maxscore
- 计算器门控（best）：min_conf=0.2, allow_task_types=[]

所有实验参数与最终配置均在 `outputs/<run_id>/config.resolved.yaml` 中可复现追溯。

---

5 实验结果

本节直接引用 Step6 自动生成的结果表与指标文件（见 `docs/TABLE_MAIN.md`、`docs/TABLE_NUMERIC.md` ），并给出关键对照结论。对应 run_id 见 `configs/step6_experiments.yaml`。

1) 检索效果（full dev / complex dev）

主表见：`docs/TABLE_MAIN.md`

关键结论（complex dev）：
baseline(post-ft) vs best multistep
- - Recall@10：0.3909465 → 0.3909465（持平）
- - MRR@10：0.2960138 → 0.2960873（+0.00007）
-
- retriever 微调带来的整体提升（full dev）：
- pre-ft baseline → post-ft baseline：Recall@10 从 0.3246 提升到 0.3772（+0.0526）

2) 数值题表现（numeric dev）

数值表见：`docs/TABLE_NUMERIC.md`

关键对照（numeric dev）：
baseline(post-ft) vs best calc gate
- - Numeric-EM：0.3838 → 0.3838（持平）
- - RelErr(mean)：683.3536 → 683.3536（持平）
- - Coverage：0.6266 → 0.6266（持平）

-
- 说明：当前版本计算器门控在 dev 上选择 `allow_task_types=[]`，以避免数值误差回退。因此 numeric 指标未出现回退，但也尚未体现提升。该结果为“安全启用”基线，可在后续提升抽取/计算置信度后再重新开启任务类型。
-
- 3) 六组矩阵实验（Step6）
- run_id 对照：
- pre_ft_baseline: `20260130_234540_ae7cdf_m01`
post_ft_baseline: `20260130_234540_ae7cdf_m02`
- post_ft_multistep_best: `20260130_234540_ae7cdf_m03`
post_ft_baseline_calc_best: `20260130_234540_ae7cdf_m04`
- post_ft_multistep_calc_best: `20260130_234540_ae7cdf_m05`
post_ft_multistep_T1_calc_best: `20260130_234540_ae7cdf_m06`
-
- 详细指标已自动写入对应的 `outputs/<run_id>/summary.json` 与 `docs/TABLE_*.md`。
-
- ---
-
- 6 错误分析与案例
-
- 本节基于 Step6 的 `error_buckets.py` 统计结果与 multistep traces，给出主要失败类型与典型案例。所有数值均可在 `outputs/<run_id>/error_bucket_stats.json` 、`outputs/<run_id>/multistep_traces.jsonl` 中复现。
-
- 1) 失败类型概览（自动统计）
-
- 以下为 Step6 六组矩阵实验的自动统计摘要：
-
- Run 20260130_234540_ae7cdf_m01: numeric_buckets={'fallback': 570}
Run 20260130_234540_ae7cdf_m02: numeric_buckets={'fallback': 570}
- Run 20260130_234540_ae7cdf_m03: complex_buckets={'max_steps': 45, 'no_gap': 525}
Run 20260130_234540_ae7cdf_m04: numeric_buckets={'fallback': 570}
- Run 20260130_234540_ae7cdf_m05: numeric_buckets={'fallback': 570}; complex_buckets={'max_steps': 45, 'no_gap': 525}
Run 20260130_234540_ae7cdf_m06: numeric_buckets={'fallback': 570}; complex_buckets={'max_steps': 46, 'no_gap': 524}
-
- 解释：
- **numeric_buckets=fallback**：由于 Step6 最优门控将 `allow_task_types=[]`，计算器任务被完全关闭，所有样本都回退到 baseline；因此 numeric 失败类型呈现为 fallback。
complex_buckets=no_gap / max_steps：多步检索在大部分样本中检测到 gap 并运行至 max_steps；在未发现 gap 的样本中直接停止（no_gap）。
-
- 2) 典型复杂查询案例（complex_dev）
-

- **qid**: `8c8c8c34`
-
- **query**:
- > Hasbro (HAS) 2023 one-time charges impact on operating profitability vs historical trends and cap allocation implications.
-
- ** 多 步 检 索 轨 迹 摘 要 **（来自 `outputs/20260130_234540_ae7cdf_m03_ms/multistep_traces.jsonl`）：
- step0: gap=MISSING_ENTITY, gap_conf=1.0, gate_decision=true, stop_reason=CONTINUE
step1: gap=MISSING_ENTITY, gap_conf=1.0, stop_reason=MAX_STEPS
- final_topk_size=10

候选证据（部分 chunk_id）：
008beea7_e0_c0
- 8c8c8c34_e0_c2
f8aec91a_e0_c1
- 8caea930_e0_c2
caa865da_e0_c3
-
- **分析**：
- 该问题涉及“对比历史趋势 + 一次性费用影响”，属于复杂查询。多步检索识别到 entity/compare 型 gap，但 refined query 与原查询高度相似，导致第二步检索未引入新证据（newly_added_chunk_ids 为空），最终以 MAX_STEPS 停止。该案例反映出 **refiner 仍偏保守**，需要进一步提升 entity 拆分与精细化 query 改写能力。
-
- 3) 数值题失败模式（numeric_dev）
-
- 当前版本中计算器通过门控被关闭（allow_task_types=[]），所有数值题回退到 baseline，从而避免 Numeric-EM 下降，但也导致 **计算器未体现增益**。后续工作需结合更可靠的单位/年份对齐与置信度校准，逐步解除 gate 并验证 Numeric-EM/RelErr 的提升。
-
- ---
-
- 7 讨论
-
- 1) 多步检索的收益与限制
- 多步检索在 complex dev 上未显著提升 Recall@10，但通过门控与合并策略避免了退化。当前的 gap 检测与 query refiner 偏规则化，**对实体拆分与对齐**仍不够稳定，导致多步检索在部分复杂查询中仅重复或轻度改写查询。
-
- 2) 计算器的可控性与覆盖率
- 数值抽取与计算模块在未充分校准前，容易出现“算得更多但算错更多”。因此 Step6 中使用 gate 将计算任务类型暂时关闭，保证 numeric 指标不回退。后续工作需要：
- 引入更鲁棒的单位与实体对齐
更细粒度的置信度打分

- 限定任务类型并改进 query-based 计算任务识别

3) 误差来源
错误分析显示，复杂查询中主要问题集中在：
gap 识别不足（无法稳定抽取比较对象）
- 合并策略未显著改善证据排序

数值问题中主要问题集中在：
结构化事实不足（缺年份或单位）
- 计算门控触发率过低

4) 可扩展方向
引入更强的 dense retriever 与领域适配（如金融领域预训练）
- 使用轻量化的 query rewriting 或规则图谱进行更精准的 gap 解析

扩展计算器支持更多指标、单位与财务结构
-
- ---
-
- 8 结论
-
- 本文构建了一个可复现的金融 RAG 工程体系，覆盖数据规范化、检索、评测与实验编排，并在此基础上实现了多步检索与显式计算模块。实验表明：
-
- 检索器微调显著提升了整体检索表现（full dev Recall@10: 0.3246 → 0.3772）。

多步检索在复杂子集上不再造成性能回退，并在 MRR 上呈现轻微提升。
- 计算器通过门控避免了数值指标退化，为后续提升奠定了稳定基线。

未来工作将聚焦于：提升 gap 识别与多步合并策略的有效性、增强数值抽取与单位对齐的鲁棒性，并探索更强的检索与推理模块。

附录 A 结果表格

表 A1 主结果（full dev 与 complex dev）

label	run_id	full_r10	full_mrr10	complex_r10	complex_mrr10
pre_ft_baseline	20260130_234540_ae7cdf_m01	0.3246	0.2030	0.3457	0.2330
post_ft_baseline	20260130_234540_ae7cdf_m02	0.3772	0.2601	0.3909	0.2960
post_ft_multistep_best	20260130_234540_ae7cdf_	0.3772	0.2601	0.3909	0.2961

	m03				
post_ft_baseline_calc_best	20260130_234540_ae7cdf_m04	0.3772	0.2601	0.3909	0.2960
post_ft_multistep_calc_best	20260130_234540_ae7cdf_m05	0.3772	0.2601	0.3909	0.2961
post_ft_multistep_T1_calc_best	20260130_234540_ae7cdf_m06	0.3772	0.2601	0.3909	0.2960

表 A2 数值题结果（numeric dev）

label	run_id	num_em	num_rel	num_cov
pre_ft_baseline	20260130_234540_ae7cdf_m01	0.3791	2874.5248	0.6202
post_ft_baseline	20260130_234540_ae7cdf_m02	0.3838	683.3536	0.6266
post_ft_multistep_best	20260130_234540_ae7cdf_m03	-	-	-
post_ft_baseline_calc_best	20260130_234540_ae7cdf_m04	0.3838	683.3536	0.6266
post_ft_multistep_calc_best	20260130_234540_ae7cdf_m05	0.3838	683.3536	0.6266
post_ft_multistep_T1_calc_best	20260130_234540_ae7cdf_m06	0.3838	683.3536	0.6266

表 A3 消融结果

label	run_id	full_r10	full_mrr10	complex_r10	complex_mrr10
post_ft_multistep_T1_calc_best	20260130_234540_ae7cdf_m06	0.3772	0.2601	0.3909	0.2960

附录 B 典型复杂查询案例

5.4 典型复杂查询案例（3 个）

案例 1（qid=8c8c8c34）
Query：Hasbro (HAS) 2023 one-time charges impact on operating profitability vs historical trends and cap allocation implications.

- Gold Answer（摘要）：In 2023, Hasbro’s operating result turned from a profit in prior years (407.7 million in 2022 and 763.3 million in 2021) to an operating loss of 1,538.8 million…

Step0 Top3：008beea7_e0_c0，8c8c8c34_e0_c2，f8aec91a_e0_c1

- Step1 Top3：008beea7_e0_c0，f8aec91a_e0_c1，8c8c8c34_e0_c2

gap/stop：MISSING_ENTITY / MAX_STEPS，final_topk_size=10

- 分析：该问题包含对比关系与年份信息，多步检索识别到 gap，但 refined query 与原查询高度相似，导致后续步骤新增证据较少，表明实体拆分与重写仍需加强。

案例 2（qid=52e25ec7）

Query：Impact on net investing cash flows from EUC sale cash inflow offsets vs acquisition outflows, AVGO.

- Gold Answer（摘要）：The $3,485 million inflow from the sale of the EUC business helped to partially offset the significantly higher cash expenditures related to acquisitions…

Step0 Top3：506e7d1e_e0_c0，52e25ec7_e0_c0，e4661352_e0_c3

- Step1 Top3：506e7d1e_e0_c0，52e25ec7_e0_c0，1c47856d_e0_c1

gap/stop：MISSING_ENTITY / MAX_STEPS，final_topk_size=10

- 分析：问题涉及“出售现金流入 vs 并购现金流出”的对比。多步检索能够维持对核心证据的覆盖，但仍未显著扩展证据范围。

案例 3（qid=ed746c33）

Query：Cash flow & cap alloc implications of IRM's ASC 842 storage rev rec vs other lines.

- Gold Answer（摘要）：For its Global Data Center Business, Iron Mountain recognizes storage revenues under ASC 842…

Step0 Top3：ed746c33_e0_c0，2a8785e8_e0_c15，a68b8600_e0_c5

- gap/stop：NO_GAP / NO_GAP，final_topk_size=10

分析：该类问题实体明确、语义集中，单步即可覆盖核心证据，多步检索不会引入额外噪声。

-
- 参考文献
- [1] 参考文献占位。
-