ChatGPT

毕业论文文字部分撰写流程 SOP

以下 SOP 基于用户提供的 WCIS24/ohyeah 项目代码和文档（包括“仓库地图.pdf”和“金融AI选题评价.pdf”），指导论文从准备到撰写各章节的全过程。所有涉及数值、指标、图表等内容，都需对应项目源码或文档中的依据，缺少时标注“TODO”并说明所需信息。

Phase 0：项目准备与证据收集

- **步骤 0.1** – 【目标】：全面了解项目背景和内容，明确论文的核心任务。
【输入】：阅读仓库地图.pdf（项目结构介绍）[1][2]、金融AI选题评价.pdf（项目指标与规划）[3][4]、README.md（项目概述）[5]。
【操作】：研读上述文档，记录数据集规模、研究周期、模块流程、评估指标等关键信息。
【产出】：项目概况笔记（含FinDER数据集规模、研究周期、流水线步骤、指标体系等）。
【验收】：完成笔记且包含：FinDER包含5703查询三元组[6]；项目周期≈2个月[7]；检索与QA管道步骤；使用Recall@5、Exact Match等指标[3][4]。
【常见坑】：忽略指标说明或模块说明；信息遗漏（需标注来源以备引用）。

- **步骤 0.2** – 【目标】：搭建运行环境并验证环境正确性。
【输入】：项目源码、requirements.txt、Makefile。[8]
【操作】：按照 README 中的设置说明创建虚拟环境并安装依赖（`pip install -r requirements.txt` 或 `make setup`）。
【产出】：可运行的项目环境。
【验收】：成功运行 `python scripts/smoke.py --config configs/smoke.yaml`，并输出 `outputs/<run_id>/metrics.json`、`logs.txt` 等文件[9][10]。
【常见坑】：依赖未安装或版本冲突；无 `make` 时改用手动命令；结果目录没有写权限。

- **步骤 0.3** – 【目标】：运行 Smoke 测试获取基线指标。
【输入】：configs/smoke.yaml、scripts/smoke.py。
【操作】：执行 `python scripts/smoke.py --config configs/smoke.yaml`，生成小规模测试结果。
【产出】：`outputs/<run_id>/metrics.json`（含 recall@5、Exact Match 等指标[11]），`logs.txt`，以及配置快照。
【验收】：metrics.json 文件存在，内容包括 `"recall@5"` 和 `"em"` 字段[11]；logs.txt 记录命令行信息；配置文件在输出目录生成。
【常见坑】：若 subset_size <=0，脚本会默认为20[12]；记得检查 `seed`（默认42）和输出目录；日志中查看 `k=5` 配置[13][14]。

- **步骤 0.4** – 【目标】：运行基线检索与问答流水线，获取主要性能数据。
【输入】：configs/prepare_data.yaml、configs/build_corpus.yaml、configs/eval_retrieval.yaml、configs/run_baseline.yaml、configs/eval_qa.yaml。[15][16]
【操作】：依次运行准备数据（normalize、切块）、构建语料、单步检索、Baseline RAG、QA 评估：如 README 所示执行准备数据、build_corpus、eval_retrieval、run_baseline、eval_qa 等脚本[15][17]。
【产出】：baseline 运行输出，包括各阶段的 metrics.json（检索、QA 精确率）、retrieval_results.jsonl、predictions.jsonl 等。
【验收】：输出目录中存在 `metrics.json`（含 Recall@K、Exact Match 等）、`retrieval_results.jsonl`、`predictions.jsonl`；运行日志成功，没有报错；指标值合理（如 recall@5≥某值）。
【常见坑】：数据集较大，内存/时间不足；参数配置遗漏；检索结果与评估脚本路径不匹配；注意使用与 Smoke 相同的随机种子与配置复现结果[14][18]。

- **步骤 0.5** – 【目标】：运行多步检索及计算器模块，收集提升效果。
【输入】：configs/build_subsets.yaml、configs/run_multistep.yaml、configs/eval_multistep.yaml、configs/run_with_calculator.yaml、configs/eval_numeric.yaml。[19][20]
【操作】：先构建测试子集 `build_subsets`，然后执行多步检索 `run_multistep_retrieval.py` 并评估，多步检索运行后接入计

算器模块 `run_with_calculator.py` 并评估数值QA。
【产出】：多步检索相关输出，包括 `multistep_traces.jsonl`（多步轨迹）、`retrieval_results.jsonl`、`predictions_calc.jsonl`、`numeric_metrics.json` 等。
【验收】：所有步骤输出文件产生；多步检索输出 recall 和 MRR 等指标；计算器输出包括事实提取和计算结果。
【常见坑】：确保使用与基线相同的 `outputs/<run_id>` 结构；注意配置文件中的 `top_k_each_step` 和 `top_k_final` 设置；若耗时，可先在较小子集上验证结果；检查是否覆盖所有复杂问题的指标 [3] [21]。

- **步骤 0.6** – 【目标】：整理实验数据与结果，形成可引用的证据。
【输入】：前几步生成的 outputs 目录中所有 `metrics.json`、`retrieval_results.jsonl`、`predictions.jsonl` 等文件。
【操作】：汇总 baseline 与多步、多阶段实验的关键指标（Recall@K、Exact Match、MRR、数值误差等）；生成对比表格或图表；记录模型配置和参数 [18]。如缺少运行数据，则标注 TODO 所需 `outputs/<run_id>` 路径或运行命令。
【产出】：实验结果汇总表（如 CSV/Markdown 表格），包含所有关键指标；生成的图表（可选）。
【验收】：对比表格正确显示各试验指标，明晰 baseline 与多步检索的改进；所有表格单元格有出处标注；若输出不全，则产生 TODO 清单。
【常见坑】：遗漏某次实验；指标计算错误（注意 Exact Match 与 recall_at_k 区别 [11]）；图表目录缺失路径引用。

- **步骤 0.7** – 【目标】：草拟论文总体框架与主要章节内容。
【输入】：项目背景资料（仓库地图、评估方案）、个人草稿、导师要求。
【操作】：根据论文常见结构，列出绪论、相关工作、方法、实验、讨论、结论、摘要等章节大纲，初步确定每章要点和安排。
【产出】：论文大纲文档，各章节标题及小节标题。
【验收】：大纲清晰完整，涵盖提到的所有章节，逻辑连贯；每章主题明确，不遗漏任何关键部分。
【常见坑】：章节遗漏或重复，逻辑跳跃；未考虑“错误分析”等必要小节；大纲过于笼统，需细化。

Phase 1：绪论（Introduction）撰写

- **步骤 1.1** – 【目标】：明确绪论结构，挖掘课题背景与动机。
【输入】：仓库地图（项目介绍）[2]、金融AI评估方案（项目背景说明）[22] [3]。
【操作】：整理引言要点：金融问答的重要性、FinDER 数据集规模 [6]、当前检索问答挑战、项目目标及贡献。
【产出】：绪论提纲（要点列表，包括数据规模、问题定义、创新之处）。
【验收】：提纲涵盖：研究领域介绍、具体问题陈述、方法概览与贡献点；引入FinDER数据集背景并引用相关数据；项目动机清晰。
【常见坑】：论述过于泛泛、缺少具体数据支撑；将方法细节或实验结果提前叙述；动机与贡献不清。

- **步骤 1.2** – 【目标】：写作绪论最小段落骨架。
【输入】：步骤1.1输出的要点。
【操作】：为绪论每一段写一句话主题句，构成“每段一句话”的框架。例如：第一段介绍金融问答背景，第二段指出传统单步检索不足，第三段提出本项目方法和贡献。
【产出】：绪论的段落主题句列表。
【验收】：每段主题句清晰、完整，逻辑连贯；覆盖前述要点；语句简洁明确。
【常见坑】：主题句冗长或含义不明；段落安排混乱，顺序不合理。

- **步骤 1.3** – 【目标】：撰写绪论正文初稿。
【输入】：步骤1.1提纲、步骤1.2主题句。
【操作】：根据框架分段落撰写，句式应准确流畅。首段介绍金融 QA 背景，引用 FinDER 数据集规模 [6]；提及评估目标和研究周期 [7]；中间段讲述项目必要性；最后总结本文贡献。
【产出】：绪论章节初稿段落。
【验收】：每段完整陈述一个主题，语言通顺；正确引用了数据规模和研究周期等信息 [22]；文章目的和贡献表述清晰。
【常见坑】：第一人称视角、学术性不足；缺少引用支持（如FinDER数据说明）；动机与方法混淆。

- **步骤 1.4** – 【目标】：插入参考证据并核对引用格式。
【输入】：绪论初稿、仓库代码及文档中的具体内容。
【操作】：在文中引用必要的项目文档和代码出处。例如，引用仓库地图中提到的数据规模 [6]、评估文档中的指标说明 [3]，以及README对评估指标的注释 [5]。验证引用格式与数量级一致。
【产出】：带有证据引用的绪论稿，如“FinDER 数据集包含 5703 个查询-证据-答案三元组

[6]”。
【验收】：所有“重点数值/参数”都有对应文档或代码引用；引用标注准确无误；内容与引用来源一致。
【常见坑】：引用不符合上下文（错误页码或行号）；未在文本中注释引用意义；遗漏必要引用导致内容无据。

- **步骤 1.5** – 【目标】：校对绪论，确保回答审稿人关切。
【输入】：带引用的绪论稿。
【操作】：根据审稿人常见疑问清单（见下文）检查绪论是否覆盖所有重点（如研究意义、研究目标、贡献总结等）。修改语句流畅性和逻辑连贯性。
【产出】：完成后的绪论文本。
【验收】：绪论包含清晰的问题陈述、研究背景、目标、贡献概述；无语法或逻辑错误；满足下文“一句话验收标准”。
【常见坑】：缺少对研究意义和贡献的总结；引用信息堆砌、不成段落；语句堆叠无主题句。

Phase 2：相关工作（Related Work）撰写

- **步骤 2.1** – 【目标】：界定相关研究领域和代表性工作。
【输入】：领域知识、金融检索QA类文献（需自行查找）。
【操作】：梳理与本项目相关的研究方向，如传统信息检索（TF-IDF、BM25）、向量检索、检索增强生成（RAG）、多跳问答以及金融语境下的数值QA等。整理文献清单及摘要。
【产出】：相关工作提纲（每个方向的主要研究和不足）。
【验收】：覆盖检索与问答、检索增强生成、金融问答方法等领域；简要说明各自特点与不足；确定需要重点引用的几篇代表性论文（在论文中用“TODO”标记）。
【常见坑】：遗漏关键领域（如矢量检索与RAG）；将自己的工作内容误认为已有工作；混淆相关工作与自己的方法。

- **步骤 2.2** – 【目标】：撰写相关工作章节草稿。
【输入】：步骤2.1的提纲和总结。
【操作】：按从一般到具体顺序分段撰写：先介绍传统检索模型和RAG框架，再谈多跳检索/问答的研究进展，最后提及金融领域相关工作。明确指出目前方法的不足和本项目的不同之处。
【产出】：相关工作章节初稿段落。
【验收】：每段覆盖一个研究方向；段落之间衔接合理；指出与本工作的差异和创新点；语言专业规范；正文中预留引用位置（以“TODO”标记，待补充文献）。
【常见坑】：参考文献缺失（需补充外部文献，不可仅靠内部文档）；与引言重复写作内容；评述宽泛，缺少深度。

- **步骤 2.3** – 【目标】：整理并格式化参考文献。
【输入】：相关工作章节草稿中标注的“TODO”引用。
【操作】：查阅学术数据库，找到并阅读需要引用的论文，补充引用（在正文使用恰当格式引用外部论文，例如：[XL著， XX会议年份]）。保证文中引用的研究与论述一致。
【产出】：更新后的相关工作章节，含完整引用。
【验收】：每个“TODO”处均已填写正确引用；引用文献格式符合学校或会议要求；章节段落完整、无“TODO”。
【常见坑】：引用不匹配实际内容；疏漏重要文献；引用格式不统一。

Phase 3：方法（Method）撰写

- **步骤 3.1** – 【目标】：概述方法流程与结构。
【输入】：仓库地图（模块结构）[23]、README中描述的流水线步骤 [16] [24]。
【操作】：根据项目模块划分，说明系统总流程：包括数据准备、单步检索、检索器训练、多步检索策略、数值计算器等模块的工作原理和交互。
【产出】：方法章节概述段落提纲。
【验收】：方案概览清晰，涵盖所有主要模块；使用示意图或表格（如可能）辅助说明；引用“仓库地图”来说明模块（如混合检索器、迭代引擎等）[23] [2]。
【常见坑】：忽略某些模块说明；逻辑跳跃（未说明各步骤之间如何衔接）；过早详述细节未提供概览。

- **步骤 3.2** – 【目标】：详细描述核心组件实现。
【输入】：项目源码（src/retrieval、src/multistep、src/calculator 等）[23]、配置文件示例。
【操作】：重点介绍检索器（TF-IDF、BM25、Hybrid Retriever 的实现），多步检索的迭代策略（engine、planner、refiner 等类的作用）[23]，以及数值问答计算模块（提取数字与同比运算）[25]。说明技术细节时引用源码或算法定义，例如SBERT 模型名称 [26]。
【产出】：组件实现描述段落。
【验收】：清晰解释每个组件的功能和工作原理；关键参数和模型（如 `max_features=20000, ngram_range=(1,2)` [27]）被说明；引用源码

文件名或行号（如 `src/retrieval/retriever.py`）；术语使用准确。
【常见坑】：描述过于口头而无结构；省略重要实现细节（如检索排序、迭代停止条件）；引用不明确或遗漏路径。

- **步骤 3.3** – 【目标】：说明配置与可复现性要求。
【输入】：AGENTS.md（可复现性规则）[18]、配置文件实例。
【操作】：说明如何使用 YAML 配置控制各模块（参考 configs/ 目录的示例），并强调设置随机种子保证可复现 [28] [18]。提及每次运行都会保存 config.yaml、metrics.json、git hash 等到 outputs [18] [2]。
【产出】：配置与复现性段落。
【验收】：说明了主要配置项及其意义；阐述了复现流程（记录 seed、commit hash 等）[18]；所有代码示例行号或文件路径正确；语言专业。
【常见坑】：忽略复现细节；使用“黑盒”概念不清楚；未说明如何设置参数、如何运行脚本即可复现。

Phase 4：实验（Experiments & Results）撰写

- **步骤 4.1** – 【目标】：介绍实验设置。
【输入】：实验数据说明（FinDER 测试集划分）、脚本配置（configs/eval_*.yaml）。
【操作】：说明实验环境与设置：使用 FinDER 全量测试集、随机种子设置、评估指标（Recall@K、Exact Match 等），并列出与基线和多步实验相关的配置文件。 [3] [29]
【产出】：实验设置段落。
【验收】：清晰列出数据集规模和分割（有无划分训练/测试）；明确运行的版本或 run_id (若有)；说明所用指标与计算方式（参考 [3] [29]）。
【常见坑】：实验设定不明确（如没有说明 K 值）；忽视统计细节（如测试集规模）；没有说明评估标准来源。

- **步骤 4.2** – 【目标】：呈现实验结果。
【输入】：从步骤0.6整理的结果表或图表。
【操作】：按照基线与多步检索模块顺序，描述各组实验结果：检索性能（Recall@K, MRR 等）、QA 准确度(Exact Match)、数值计算误差（若有）。引用具体数值或表格，并说明相对提升。表格或图表应列出关键信息（如 baseline vs 我们的方法）。若缺少某些结果数据，则标注 TODO 输出文件路径，例如 `outputs/<run_id>/metrics.json`。
【产出】：实验结果段落（可插入表格/引用具体数字）。
【验收】：对比分析明确：多步检索提升了复杂查询 Recall@10、EM等指标；数值任务准确率；表格或图表与文本对应；引用所有数字的来源（如“见表X中的Recall@5”）。
【常见坑】：抄写结果时不引用源；结果数值与源不符；过度解读（应实事求是）。

- **步骤 4.3** – 【目标】：补充实验分析（举例与错误分析）。
【输入】：Multistep_traces.jsonl 中的典型案例、错误分析脚本 outputs（若有）。
【操作】：选择几个典型复杂问题，对比基线系统和本系统的检索链路和答案；描述模型如何通过多步策略弥补遗漏。简要提及常见错误类型（检索漏召回、推理漏步、生成偏差），并说明改进方向 [30]。
【产出】：典型案例分析段落、错误分类说明（可作为表格或列表形式）。
【验收】：至少包含一个案例说明（问题、参考答案、基线结果、本系统结果）；错误类型分类清晰（如召回不足、计算错误、表达瑕疵）[30]；提供针对性讨论。
【常见坑】：只罗列数字缺乏解释；案例分析过于简单；与前面结果重复。

Phase 5：讨论（Discussion）撰写

- **步骤 5.1** – 【目标】：综合分析结果，探讨方法优势与局限。
【输入】：前面实验结果与案例分析输出。
【操作】：讨论多步检索在复杂查询上的优势（结合结果说明）和可能的开销，反思误差案例原因；分析数值计算模块在什么场景下有效。结合“金融AI评估”中对效率的关注，说明时间复杂度与可扩展性 [31]。
【产出】：讨论章节段落。
【验收】：论述涵盖：多步检索优势、误差来源、系统效率；不做无证据的夸大；论点与实验数据对应；引用评估方案中的效率关注 [31]。
【常见坑】：仅重复结果；忽视实验目标；忽略未来改进方向；逻辑跳跃。

- **步骤 5.2** – 【目标】：指出研究局限和未来工作。
【输入】：项目时间和资源限制（2个月开发时间），评估方案关注点 [32] [31]。
【操作】：讨论在2个月内未能深入优化的部分（如更多召回策略、实时性问题），结合评估文档中提到的“效率指标”等说明后续可探索方向 [31] [33]。
【产出】：局限与未来工作段落。
【验收】：提及具体局限（时间、硬件、数据量、模块能力），描述

可行的改进措施；语言客观。
【常见坑】：避重就轻或自吹自擂；未提到任何局限；未给出明确改进建议。

Phase 6：结论（Conclusion）撰写

- **步骤 6.1** – 【目标】：总结全文并强调贡献。
【输入】：前文各章结论性观点和实验成果。
【操作】：撰写结论段落：简述研究目的，重复核心结果（如提高了复杂查询的EM/Recall）和主要贡献（多步检索框架、数值QA集成）。避免引入新信息，只总结要点。
【产出】：结论章节段落。
【验收】：结论明确概括了论文贡献和实验验证结果，与引言相呼应；语言简练；无“TODO”内容。
【常见坑】：简单重复引言内容；介绍新方法或数据；结论空洞无新信息。

Phase 7：摘要（Abstract）撰写

- **步骤 7.1** – 【目标】：撰写论文摘要，浓缩核心内容。
【输入】：论文各章要点（问题、方法、结果、贡献）。
【操作】：按照“背景-问题-方法-结果-贡献”的顺序，用4~6句话概括整篇论文：1）研究背景与问题；2）提出的方法；3）主要实验结果（如提升了Recall@K/EM）；4）结论与贡献。摘要中不引用具体文献，但可引用实数（如5703个样本）。
【产出】：摘要文本。
【验收】：摘要篇幅适中，涵盖了研究动机、方法概述、关键结果和贡献；数据描述准确（引用数据集规模 6 ），语言凝练；无需参考文献即可独立阅读。
【常见坑】：冗长或漏重要信息；未体现实验成果；叙述不一致。

Phase 8：终稿整理与校对

- **步骤 8.1** – 【目标】：检查整篇论文的一致性和格式。
【输入】：初稿全文。
【操作】：逐章校对：检查逻辑衔接、术语统一、图表与引用一致；确保图表、表格路径正确；参考文献格式无误。使用文档审校工具（如词频检查、拼写检查）。
【产出】：无语法和格式错误的最终稿。
【验收】：全文结构清晰完整，段落标题正确，所有引用和链接可定位；符合毕业论文格式要求；无待办TODO。
【常见坑】：遗漏“TODO”未解决；格式细节不符（如图表题注、参考文献样式）；章节名称或编号错误。

- **步骤 8.2** – 【目标】：写作任务单输出与自查。
【输入】：前述章节任务单要求。
【操作】：整理各章节的“审稿人关心的问题”、“段落骨架”等任务单内容（见下文）。逐项对照检查是否满足验收标准。
【产出】：完成的写作任务单（紧接本文后列出）。
【验收】：每章节任务单齐全无缺，摘要符合上文要求；确认所有阶段和任务完成，论文质量符合验收标准。
【常见坑】：任务单内容与正文不符；逻辑跳跃或遗漏关键项目。

各章节写作任务单

绪论

- **审稿人关心的问题清单**：研究的金融问答背景与意义是什么？为什么需要多步检索？FinDER数据集有哪些特点？本文的研究目标和贡献是什么？采用哪些数据和评估指标？
- **最小段落骨架**：
- “金融问答系统近年来得到关注，针对复杂查询需要可靠的信息检索与推理，FinDER 数据集包含 5703 条查询-证据-答案三元组 6 。”
- “传统一次检索方法在多跳查询上召回不足，难以整合多段证据满足推理需求。”
- “本文提出结合多步迭代检索和计算模块的新框架，实现对复杂金融查询更全面的检索和计算推理。”
- “实验在 FinDER 数据集上验证了方法有效性，较基线提高了Recall@K和Exact Match等指标。”

- **证据文件路径清单**：金融AI选题评价.pdf（第1页）；仓库地图.pdf（第2页有关数据集规模和项目周期）；README.md（项目目标和指标说明） [5] ；项目代码（说明多步检索与计算模块） [23] 。
- **常见逻辑错误**：背景介绍太宽泛，缺少数字或文献支撑；将过多技术细节提前；未明确研究目的或贡献；动机表述不清。
- **完成后的一句话验收标准**：绪论清晰交待研究背景、问题和贡献，引用了项目中的数据和指标，满足下文给出的验收标准。

相关工作

- **审稿人关心的问题清单**：相关领域有哪些研究工作？多步检索与传统检索方法有何区别？金融问答领域有没有现有方法？本文工作与现有研究相比创新点何在？
- **最小段落骨架**：
- “经典信息检索方法（如TF-IDF和BM25）和RAG模型被广泛用于QA任务，但在多跳推理需求下召回有限。”
- “近年来，研究者提出了多步检索和链式推理技术，以改善复杂查询的召回率和答案准确率。”
- “金融领域问答相对较少，现有工作主要关注金融文本分析和简单问答，本研究填补了基于金融复杂检索的空白。”
- “与相关方法相比，本方法引入了计算模块和迭代检索设计，有望在评估指标上超越基线系统。”
- **证据文件路径清单**：暂无内部文件直接讨论此内容（需要补充外部文献）；可参考金融AI选题评价中的动机描述。
- **常见逻辑错误**：未涵盖主要相关方法（如只提简单检索模型）；将方法细节或结果提前到相关工作；与自己工作对比分析薄弱。
- **完成后的一句话验收标准**：相关工作章节准确回顾了领域内主要研究，明确了方法演进和本文创新点，且引用了关键文献。

方法

- **审稿人关心的问题清单**：系统整体架构如何？各模块（检索、多步检索、数值计算等）如何实现？使用了哪些模型或参数？配置如何组织？如何保证实验可复现？
- **最小段落骨架**：
- “整个系统包含数据预处理、单步检索、多步迭代检索和数值计算四个模块。”
- “单步检索使用TF-IDF和BM25，同时支持HybridRetriever（BM25+SBERT） [26] 。”
- “多步检索模块利用 planner 和 refiner 等组件迭代提取补充证据，直到满足条件。”
- “数值计算模块提取文本中的数值并执行同比增长、差值等运算 [25] 。”
- “系统所有参数通过 YAML 配置控制，运行时记录配置快照、随机种子和 Git 提交哈希以保证可复现 [18] 。”
- **证据文件路径清单**：src/retrieval/retriever.py；src/multistep/engine.py、planner.py、refiner.py；src/calculator/extract.py、compute.py；AGENTS.md [18] ；configs/*.yaml (示例配置)；仓库地图.pdf（说明各模块功能） [23] [2] 。
- **常见逻辑错误**：对模块实现只字不提代码或具体算法；术语和符号不统一；遗漏重要参数设置（如检索K值）；可复现细节缺失。
- **完成后的一句话验收标准**：方法章节完整描述了系统架构和关键算法实现，并引用了相应源码或配置，读者可理解系统设计与运行流程。

实验与结果

- **审稿人关心的问题清单**：实验设置是什么？数据集规模和分割如何？使用了哪些评估指标？baseline与本方法各自的结果如何？提升在哪些方面显著？是否有典型案例说明？
- **最小段落骨架**：
- “我们在 FinDER 数据集的测试集上评估系统性能（使用Recall@K和Exact Match指标 [3] [29] ）。实验使用相同的随机种子和预处理设置。”

- “基线单步检索结果：Recall@5 = X%（EM = Y%），多步检索后对应指标提高至 A%（B%），见表格1。”
- “在数值问答任务上，集成计算模块后Exact Match提升Z点，减少了数值误差（详见表格2）。该结果证明了本方法对定量查询的改进。”
- “典型案例分析：对于复杂查询 Q，基线未检索到所有证据，多步检索成功找全，获得正确答案；详见图3所示检索轨迹和答案对比。”
- **证据文件路径清单**：outputs/<baseline_run_id>/metrics.json、outputs/<baseline_run_id>/retrieval_results.jsonl、outputs/<baseline_run_id>/predictions.jsonl；outputs/<multistep_run_id>/metrics.json、retrieval_results.jsonl、predictions_calc.jsonl；（如缺少请运行相应脚本或记录 TODO）；金融AI选题评价.pdf（指标定义） [3] [29]；评估脚本 configs/eval_*.yaml。
- **常见逻辑错误**：结果描述未引用具体数字来源；只叙述改进未说明实际数值；忽略基线对比；将图表标题或数值写错；案例说明与数据不符。
- **完成后的一句话验收标准**：实验章节完整展示了所有关键指标对比，数据来源明确（表格/引用），说明了本方法相对于基线的具体提升。

讨论

- **审稿人关心的问题清单**：结果说明了什么？为何多步检索能提升性能？还存在哪些问题？效率如何？系统在真实环境下的潜在应用或改进方向是什么？
- **最小段落骨架**：
- “结果表明，多步检索在复杂查询中显著提高了证据召回率，原因在于迭代检索补充了基线遗漏的证据。”
- “集成计算模块使得定量查询得到更精确答案，减少了错误率；但对于部分文本问答提升有限。”
- “本系统采用多次检索与LLM计算，增加了响应时间；评估表明平均查询耗时 X 秒（满足交互式需求要求 [31]），未来需优化效率。”
- “我们分析了检索和推理过程中的典型错误类型，如召回不足、推理链断裂、答案表述不当，并讨论了针对性改进方案。”
- **证据文件路径清单**：错误分析脚本输出（若有），多步检索输出文件（multistep_traces.jsonl）；金融AI选题评价.pdf（效率和错误分析建议） [30] [31]；相关部件配置（计算步数上限等）。
- **常见逻辑错误**：过度推测结果意义；忽视负面结果；未结合实际指标说明因果；遗漏讨论实验限制和改进空间。
- **完成后的一句话验收标准**：讨论章节深入分析了实验结果的意义和原因，指出了系统的局限与改进方向，论点与数据相符。

结论

- **审稿人关心的问题清单**：本文实现了什么？主要发现与贡献是什么？对未来研究有什么启示？
- **最小段落骨架**：
- “本文提出了一种面向金融多步检索的RAG框架，并引入数值计算模块以解决复杂金融问答。”
- “在 FinDER 数据集上进行的实验验证了所提方法的有效性：与基线相比，复杂查询的召回率和准确率有显著提升。”
- “总结来说，我们的方法证明了多步检索与显式计算的优势，为金融QA系统的设计提供了新思路。”
- **证据文件路径清单**：无直接引用（结论可概括前文提到的结果，无需新证据）。
- **常见逻辑错误**：未明确总结本研究成果；引入新信息；与前文不呼应。
- **完成后的一句话验收标准**：结论简明扼要地总结了研究目标、主要结果和贡献，与论文整体相呼应。

摘要

- **审稿人关心的问题清单**：研究背景是什么？解决了什么问题？使用了哪些方法？取得了什么效果？贡献点有哪些？
- **最小段落骨架**：

- “金融问答的复杂多跳查询亟需高效检索与推理方法，本研究基于 FinDER 数据集开展工作。”
- “我们提出一个多步检索加计算器的框架，使系统能够迭代检索补充证据并进行定量计算。”
- “在实验中，该方法相比传统基线提高了复杂查询的Recall@K和Exact Match等指标。”
- “该研究为复杂金融问答问题的解决提供了新的思路和参考。”
- **证据文件路径清单**：引用仓库地图中的数据集规模 6 和项目周期 7；引用评估方案中的指标定义或本章结果（若需要）。
- **常见逻辑错误**：摘要过长或过短；遗漏主要结果；无具体数据支撑论断；结构混乱。
- **完成后的一句话验收标准**：摘要简洁概述了研究问题、方法和结果，并突出关键数据指标，使读者能够快速了解论文贡献。

---

1 2 6 7 22 23 25 26 27 仓库地图.pdf

file://file_0000000053647208ad8448982d1aef10

3 4 21 29 30 31 32 33 金融AI选题评价.pdf

file://file_00000000ef607208995842b7b94b8447

5 8 9 15 16 17 19 20 24 README.md

https://github.com/WCIS24/ohyeah/blob/f37c228262dcd5747072a9b202426de06f95e872/README.md

10 11 12 13 14 28 smoke.py

https://github.com/WCIS24/ohyeah/blob/f37c228262dcd5747072a9b202426de06f95e872/scripts/smoke.py

18 AGENTS.md

https://github.com/WCIS24/ohyeah/blob/f37c228262dcd5747072a9b202426de06f95e872/AGENTS.md