附件 3

场景挑战赛道赛题及竞赛规则
金融多模态知识库构建与复杂问答检索算法

一、赛题背景

随着金融行业的数字化转型加速，海量的非结构化文档（如研究报告、合同、财报、政策文件、演示文稿等）构成了金融机构的核心知识资产。如何从这些多模态、多格式的文档中，快速、精准地提取关键信息，并智能地回答复杂的用户问题，已成为提升投研效率、风险控制能力和客户服务水平的关键技术。

传统的关键词匹配检索技术已难以应对金融领域多意图、跨文档、推理类的复杂查询需求。本次竞赛旨在探索和推动新一代智能检索与问答技术的发展，聚焦于对复杂问题的深度意图理解和精准知识碎片采编，打造更懂金融、更懂用户的智能知识大脑，为行业提供技术储备和人才选拔平台。

二、赛题应用场景

1. 客户服务与营销支持：在银行客服热线、网上银行客服、手机银行客服等场景中，客户会提出各类咨询问题，如理财产品推荐、贷款申请条件等。客服人员能够迅速检索到相关知识点，为客户提供及时、准确的解答，提升客户满意度。

2. 风险控制与合规管理：银行风险控制和合规管理部门需要及时了解各类监管政策、行业规范以及内部风险管理制度，以便对业务进行风险评估和合规检查。相关人员需能够快速检索到所需的监管条款、风险案例等知识点，为风险控制和合规管理工作提供有力支持。

3. 内部知识管理：员工在办理信贷、理财、结算等业务时，可能会遇到政策解读、业务流程疑问、产品细节查询等问题。通过知识库采编检索算法，员工可快速输入问题，获取准确的知识点，提高业务办理效率和准确性。

三、赛题任务

参赛队伍需开发一套针对金融领域多类型文档的知识库采编检索算法系统，该系统需完成以下任务：

1. **文档解析与知识提取：**对提供的金融领域多种格式文档（word、pdf、excel、ppt、txt、markdown、png、jpeg 等）进行解析，提取文档中的关键信息和知识点，构建结构化的知识库。其中，对于 pdf、ppt 等复杂板式文档需要进行板式解析、对于 PPT、png、jpeg 等图像格式文档，需先进行图像识别与文字提取。解析完成后再进行知识提取；对于 excel 格式文档，需提取表格中的数据信息及相关说明文字作为知识点。

2. **问题意图理解：**准确理解用户输入的各类问题（包括多意图、推理、细节、

长文本、总结等）的意图，明确用户所需知识点的范围和核心需求。例如，对于多意图问题“请介绍个人住房贷款的申请流程以及最新的利率政策”，需准确识别出用户同时关注申请流程和利率政策两个意图；对于推理问题 “已知客户 A 的月收入为 8000 元，负债每月 2000 元，名下有一套价值 100 万元的房产，请问该客户申请 50 万元的信用贷款是否符合条件”，需根据知识库中的信贷政策知识点进行推理分析。

3. **知识点检索与排序：**根据用户问题的意图，从构建的知识库中检索相关的知识点，并按照与问题的相关性、准确性等因素进行排序，每个问题输出 top3 的知识点，且每个知识点的字数不超过 1500 字。对于多跳问题，需进行多步检索和关联分析，逐步获取所需知识点；对于总结类问题，需对检索到的相关知识点进行归纳总结，形成简洁、全面的回答知识点。

四、数据集及数据说明

（一）数据来源

本次竞赛数据集均来源于银行真实业务场景中的公开资料、内部合规整理的非敏感业务文档及模拟生成的符合金融业务逻辑的文档和问题，确保数据的合法性、真实性和可用性，不包含客户隐私信息、商业机密等敏感数据。

（二）数据规模

文档库：包含约 1000 份金融领域文档，格式包括：

PDF（约 300 份）：包括国家及地方金融监管部门发布的政策法规（如《商业银行法》《贷款通则》）、行业研究报告（如金融市场分析报告、行业风险评估报告）、银行内部审计报告等。


中国银行 BANK OF CHINA 研究院

全球银行业展望报告

2023 年第 4 季度（总第 56 期） 报告日期：2023 年 10 月 8 日

要点

● 2023 年以来，全球银行业盈利水平普遍提升，但规模小幅收缩，资产质量存在劣变迹象。相对而言，中国银行业支持实体经济力度不减，规模持续增长，盈利增速放缓，资产质量保持稳定。

● 展望四季度，全球银行业积极应对本国息差环境，盈利能力保持稳健，但资产质量波动更加明显。中国宏观经济稳健复苏，银行业在低息差环境下盈利保持稳定，规模扩张态势不变，资产质量向好。

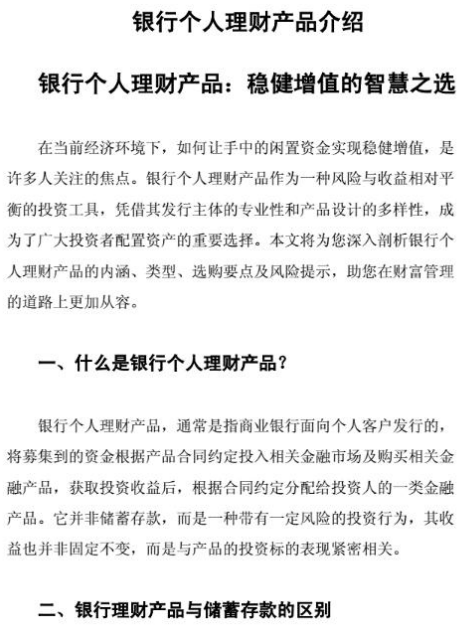
● 2023 年，全球银行业发展环境持续变化，本次季报围绕千家行榜单、不同息差环境下各国银行业应对、中国

中国银行研究院
全球银行业研究课题组

组 长：陈卫东
副组长：王家强
成 员：邵 科
叶怀斌
杜 阳
李一帆
马天娇
李 彧（香港）

Word（约 350 份）：涵盖银行各类业务管理办法、业务操作手册、产品说明书

（如理财产品说明书、贷款产品介绍）等，内容涉及信贷业务、理财业务、结算业务、国际业务等多个领域。



银行个人理财产品介绍

银行个人理财产品：稳健增值的智慧之选

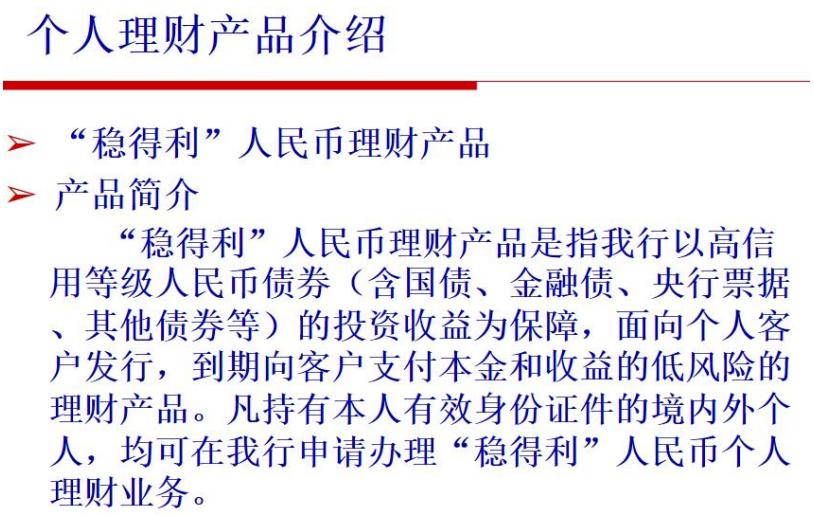
在当前经济环境下，如何让手中的闲置资金实现稳健增值，是许多人关注的焦点。银行个人理财产品作为一种风险与收益相对平衡的投资工具，凭借其发行主体的专业性和产品设计的多样性，成为了广大投资者配置资产的重要选择。本文将为您深入剖析银行个人理财产品的内涵、类型、选购要点及风险提示，助您在财富管理的道路上更加从容。

一、什么是银行个人理财产品？

银行个人理财产品，通常是指商业银行面向个人客户发行的，将募集到的资金根据产品合同约定投入相关金融市场及购买相关金融产品，获取投资收益后，根据合同约定分配给投资人的一类金融产品。它并非储蓄存款，而是一种带有一定风险的投资行为，其收益也并非固定不变，而是与产品的投资标的表现紧密相关。

二、银行理财产品与储蓄存款的区别

PPT（约 100 份）：主要是银行产品推广演示文稿（如新型理财产品推广 PPT、信贷产品介绍 PPT）、业务培训课件（如员工业务操作培训 PPT、合规知识培训 PPT）、会议汇报材料（如季度业务汇报 PPT、年度工作总结 PPT）等，包含文字、图表、图片等内容。



个人理财产品介绍

➢ “稳得利”人民币理财产品

➢ 产品简介

“稳得利”人民币理财产品是指我行以高信用等级人民币债券（含国债、金融债、央行票据、其他债券等）的投资收益为保障，面向个人客户发行，到期向客户支付本金和收益的低风险的理财产品。凡持有本人有效身份证件的境内外个人，均可在我行申请办理“稳得利”人民币个人理财业务。

Excel（约 100 份）：包含银行各类业务数据报表（如月度信贷投放数据报表、理财产品销售数据报表）、客户信息数据表（如客户基本信息表、客户资产负债数据表）、财务数据统计表（如银行年度财务报表、分支机构利润数据表）等，表格中包含数据及相关字段说明文字。

TXT（约 50 份）：包括客户咨询记录（如客服与客户的对话记录、客户留言记

录）、员工工作笔记（如业务办理笔记、问题解决记录）、内部通知公告（如业务调整通知、系统升级公告）等。

Markdown 文档（约 50 份）：主要是银行技术文档（如系统开发文档、接口说明文档）、项目计划文档（如产品研发计划、系统升级计划）等，采用 markdown 格式编写，包含标题、列表、代码块、链接等元素。

PNG/JPEG（约 50 份）：包括金融数据图表（如业务增长趋势图、客户分布统计图）、业务流程示意图（如贷款审批流程图、转账操作示意图）、产品宣传图片（如理财产品宣传海报、银行服务场景图片）等，图像中包含文字信息。

（三）测试集

包含 500 个不同类型的用户问题，问题类型涵盖多意图（75 个）、推理（75 个）、多跳（75 个）、细节（75 个）、长文本（150 个）、总结（100 个），不提供标准答案知识点，用于最终评估参赛队伍算法系统的性能。本次比赛采用两轮制，每轮问题分配为：100 / 400。

（四）数据集限制

竞赛仅允许使用比赛提供数据集。为了维护竞赛公平性，严禁使用其他公开或私有的采编检索数据集。所有数据仅限本次竞赛使用，严禁任何形式的下载、分发、泄露或用于其他商业及研究目的。

五、算法设计要求

（一）总则

目的：为确保竞赛的公平、公正、可复现性，并保护各方知识产权。

适用范围：适用于所有参赛队伍。

重要日期：请密切关注竞赛平台公告的提交截止时间，逾期提交将视为自动放弃。

（二）提交材料通用规范

1. 目录层级结构：

所有提交文件需按以下层级存放：

参赛队伍名称（根目录）/

├── 材料类型（子目录 1）/

│ └── 具体文件（按命名规则命名）

├── 材料类型（子目录 2）/

│ └── 具体文件（按命名规则命名）

└── ...（其他材料类型子目录）

• 根目录：必须以参赛队伍名称命名（如“星火队”“领航队”），用于统一收纳该队伍的所有阶段提交材料。

• 子目录：在根目录下，按 “材料类型” 创建子目录（如“技术方案”“模型

文件”“测试集检索结果”等），名称需与文件命名中的“材料类型” 一致，用于分类存放对应类型的文件。

示例：

星火队（根目录）/

├─ 技术方案/

│ └─ 技术方案.pdf

├─ 模型/

│ └─ 模型.zip

└─ 测试集检索结果/

└─result.json（压缩成 zip 文件提交）

2. 格式要求：文本类材料（技术方案、说明文档）统一为 PDF 格式，表格类材料统一为 Excel 格式，压缩包（模型、数据、代码）统一为 ZIP 格式（禁止使用 RAR 格式），且单个压缩包大小不超过 20GB（超出需分卷压缩并标注序号）。

3. 版本要求：代码及模型需兼容 Python 3.8 及以上版本，依赖库需在 requirements.txt 文件中明确版本号；文档解析需支持竞赛指定的所有格式（word/pdf/excel/ppt/txt/markdown/png/jpeg），其中图像类文件需适配主流 OCR 工具接口。

4. 合规要求：提交材料需符合《数据安全法》《个人信息保护法》及金融行业合规规定，禁止包含敏感信息（如真实客户姓名、账号、身份证号）；若涉及侵权，由参赛队伍承担全部责任。

（三）分阶段提交材料明细

1. 初赛提交材料

本阶段旨在评估算法模型的最终输出效果。

1）提交内容：result.json，压缩成 zip 文件提交。

2）文件格式：

• 必须为标准的 JSON 文件。

• 编码必须是 UTF-8。

• 顶层结构必须为 JSON 数组，数组中的每一项是独立对象，每个对象仅包含 1 个 question_id 和其对应的 knowledge_points 数组。

• 每个对象的结构要求：

question_id：字符串类型，对应单个问题的唯一标识（如 "Q20231001001"）。

knowledge_points：数组类型，每个元素是字符串格式的知识点文本

• 示例：

```
[
```

```
    {
        "question_id": "Q20231001001",
        "knowledge_points": [
          "知识点 1 的文本内容...",
          "知识点 2 的文本内容...",
          "知识点 3 的文本内容..."
        ]
    },
    {
        "question_id": "Q20231001002",
        "knowledge_points": [
          "知识点 1 的文本内容...",
          "知识点 2 的文本内容...",
          "知识点 3 的文本内容..."
        ]
    },
  {
        "question_id": "Q20231001003",
        "knowledge_points": [
          "知识点 1 的文本内容...",
          "知识点 2 的文本内容...",
          "知识点 3 的文本内容..."
        ]
  }
]
```


• 对于测试集中的所有问题，都必须提供答案。

3）提交方式：

• 通过竞赛平台的“提交结果”页面上传。

• 每人/每天不超过 2 次提交机会。

4）命名规范：文件必须命名为 result.json（压缩成 zip 文件提交）。

5）注意事项：

• 请确保 JSON 格式完全正确，否则可能导致解析失败和无成绩。

• 初赛阶段无需且不应提交任何代码或模型。

2. **决赛提交材料**

本阶段针对初赛排名前列（如 Top30）的队伍，除模型的输出结果外还需提交完整材料以供复核与评审。

1）提交内容清单：

- 模型输出结果，格式同初赛 (result.json)
- 技术方案报告 (report.pdf)
- 源代码与依赖文件(压缩包 code.zip)
- 模型文件(提供下载脚本或网盘链接)
- Docker 镜像(导出为.tar 文件)
- 其他辅助材料(可选)

2）详细要求：

2.1 技术方案报告(report.pdf)

- 格式： PDF 文
- 内容： 应包含但不限于：

①摘要与引言

②系统整体架构图

③核心模块详解（文档解析、嵌入模型、检索策略、重排序模型等）

④实验分析（训练数据使用、消融实验、效率分析）

⑤总结与展望

⑥要求：内容真实、逻辑清晰，准确反映所提交的代码实现

2.2 源代码与依赖文件 (code.zip)

- 内容：全部源代码、脚本、配置文件
- 必须包含的文件：

①README.md: 详细的说明文档，包括环境配置、训练（如果适用）、索引构建、服务启动的完整命令；

②requirements.txt 或 environment.yml: 列出所有 Python 依赖包及其精确版本；

③preprocess.py / build_index.py: 数据预处理和构建索引的脚本；

④api_server.py 或 main.py: 启动检索服务的应用程序入口脚本。

- 要求：代码结构清晰，注释完整，严格遵循 README.md 中的说明即可完成环境搭建和运行

2.3 模型文件

- 要求：为避免传输超大文件，强制要求使用下载脚本
- 方式：在代码目录 code.zip 中提供一个名为 download_models.sh 的脚本
- 脚本内容：该脚本应能自动从我们指定的云存储下载所有所需的预训练模型和

微调后的模型权重

• 备选方案：如模型无法公开访问，请在报告中标明，并与主办方联系通过私有网盘链接提交

2.4 Docker 镜像 (submission.tar)

• 目的：确保环境一致性，方便一键复现

• 要求：

①基于一个官方基础镜像（如 python:3.10-slim）

②在镜像内预装所有依赖

③将代码和模型下载脚本复制到镜像内

④暴露服务运行的端口（如 8000）

⑤设置默认启动命令为运行服务

• 提交方式：

①使用 docker save -o submission.tar <your-image-name>:<tag> 命令将镜像导出为 .tar 文件。

②上传至主办方提供的云存储链接

2.5 其他辅助材料

• 例如：特殊的配置文件等

3）提交方式：

• 技术报告、源代码压缩包、Docker 镜像需上传至主办方提供的提交入口

• 模型文件通过下载脚本集成在代码中

（四）复现与验证

主办方评审团将按以下流程验证提交的作品：

（1）环境准备：加载 Docker 镜像 (docker load -i submission.tar) 并运行容器。

（2）模型下载：在容器内执行 download_models.sh 脚本获取模型。

（3）索引构建：运行 build_index.sh 脚本，在容器内为测试知识库构建索引。

（4）服务启动：运行 run_service.sh 启动检索 API 服务。

（5）结果验证：使用一个未公开的验证问题集对运行的服务进行查询，将返回结果与初赛提交的 result.json 进行比对，验证一致性和性能。

（6）报告评审：结合技术方案报告，评估方案的创新性、完整性和工程价值。

（五）违规处理

如出现以下情况，将视为违规，并取消参赛或获奖资格：

1. 提交的答案或代码侵犯他人知识产权。
2. 技术报告与代码实现严重不符。
3. 无法在提供的 Docker 环境中成功复现系统。

4. 复现结果与初赛提交答案存在不可合理解释的显著差异。

5. 使用非规定方式（如人工标注）生成测试集答案。

（六）知识产权

1. 参赛队伍与出题方共享提交的代码和技术的知识产权。

2. 参赛队伍授予主办方出于评审、宣传竞赛成果目的的使用权。

3. 严禁对外泄露或分发主办方提供的任何数据集。

4. 请各参赛队伍严格遵守以上要求，预祝取得优异成绩！

六、性能指标要求

查询响应延迟

定义：从系统接收到一个用户问题开始，到返个知识点结束所花费的平均时间。

测量方式：在评测服务器上，使用统一的硬件环境，对测试集中的所有问题依次执行并计算平均耗时。

重要性：直接决定了用户体验。在实际应用中，秒级的响应是基本要求，亚秒级响应是理想目标。

七、效果指标要求

（一）客观量化指标：检索正确率

计算公式：Top3 检索正确率 =(检索成功的问题数量) / (总测试问题数量) * 100%

注 1：检索只要其返回的 Top3 知识点中有一个及以上被评估为“正确”，即为检索正确。

注 2：知识点必须展示原文信息，长度不能超多 1500 个字符（表格类型去除结构字符），如有需要可以携带图片

（二）主观质量指标

侧重于内容层面的深度和质量，通常需要领域专家进行评估。

1. 完整性

知识点的内容是否涵盖了文档的所有核心要点和关键细节，是否存在重要信息的缺失。

2. 一致性

知识点自身逻辑是否一致，与文档原文的上下文含义是否一致，与其他可靠信源的信息是否一致。

3. 可读性与流畅性

对于需要重组、概括的采编内容（如摘要），其语言是否通顺、逻辑是否清晰、易于理解。

4. 相关性

采编的信息是否与预设的主题或任务高度相关，是否过滤了冗余和无关信息。

八、开发环境

参赛者需要基于 Python 语言开发，可以使用开源的算法框架如 Pytorch，TensorFlow 等。

为保障竞赛公平性及避免后续商业纠纷，参赛团队所使用的基座模型需满足以下要求之一：

1. **完全自研模型：**需在技术方案报告中详细说明模型的架构设计、训练数据来源（仅限本次竞赛提供数据集）、训练过程及性能验证结果，且需提供完整的模型训练代码供主办方查验。

2. **开源模型二次开发：**需基于遵循 MIT 协议、Apache 2.0 协议等开源许可协议的模型进行优化（如 FinBERT、金融 BERT-wwm 等），且需在技术方案报告中明确标注基座模型的原始来源、开源协议类型及二次开发的具体修改内容（如模型微调策略、参数调整范围、新增模块设计等）。

3. 严禁使用非开源模型、开源协议限制商业使用的模型或未明确授权的第三方模型，若经核查发现违规使用，将取消参赛资格及成绩。

九、成绩评价

（一）第一轮：初赛 (线上)

1. 测试集: 100 个问题。
2. 评分: 100% 客观量化评分（Top-3 检索正确率）。
3. 目的: 筛选队伍进入决赛。

（二）第二轮：决赛 (综合)

1. 测试集: 400 个问题。
2. 评分: 最终成绩 =60% 客观量化评分 +20% 性能指标评分 +20% 答辩评分。

（1）客观量化评分: 基于 400 个问题的 Top-3 检索正确率。

（2）性能指标评分: 评估查询响应延迟。

（3）答辩评分: 由专家根据方案创新性、工程实现、Demo 演示和现场问答综合评定。

具体计算逻辑如下：

1）客观量化评分：基于 “Top3 检索正确率” 计算，原始得分按线性标准化方式转换为 0-100 分区间，计算公式为：标准化客观分 = （参赛队正确率 - 最低正确率）/（最高正确率 - 最低正确率）×100（若所有队伍正确率相同，则均计为 80 分）。

2）性能指标评分：以 “查询响应延迟” 为核心指标，设置基础阈值（秒级响应，即≤3 秒）与理想阈值（亚秒级响应，即≤0.5 秒）。原始延迟按以下规则

转换为 0-100 分区间：

- 延迟≤0.5 秒：计 100 分；
- 0.5 秒＜延迟≤3 秒：按线性公式计算得分，即得分 =100 - （延迟 - 0.5）/（3 - 0.5）×80；
- 3 秒＜延迟≤5 秒：计 20 分；
- 延迟＞5 秒：计 0 分。

同时，采用异常值过滤机制：若某参赛队的延迟数据超出所有队伍延迟均值的 3 倍标准差，该数据将被标记为异常值，取均值的 1.5 倍标准差作为其延迟计算值，保障评分不受极端数据干扰。

3. 主观质量评分：由 3 名及以上专家组成评审组，从 “完整性”“一致性”“可读性与流畅性”“相关性” 四个维度（各占 25% 权重）对知识点质量进行打分，取平均分作为最终主观得分（0-100 分区间）。

4. 最终总成绩 = 客观量化评分 ×60%+ 性能指标评分 ×20%+ 主观质量评分 ×20%，按总成绩从高到低排序确定排名。

十、解题思路

参赛团队需围绕 “文档解析 - 知识构建 - 意图理解 - 知识点检索” 四大核心环节设计解题方案，具体思路框架如下：

（一）文档解析与知识提取环节

针对金融领域多格式文档，需先完成格式适配与内容提取，再进行结构化知识转化，具体步骤如下：

1. 多格式文档解析

对于 txt、markdown 等纯文本文档，可采用 Python 的 txtreader、markdown-it-py 等工具直接读取文本内容，去除冗余空格、特殊符号，进行分句与分词处理；

（1）对于 word（.doc/.docx）文档，使用 python-docx 库提取文本、表格与图片信息，其中表格内容需转化为 “表头 - 行数据” 的结构化格式，图片暂存待后续处理；

（2）对于 pdf 文档，若为可复制文本型 pdf，采用 PyPDF2 或 pdfplumber 提取文本与表格；若为扫描型 pdf，需先通过 OCR 工具（如 Tesseract-OCR、百度智能云 OCR）将图像转化为文本，再进行内容提取，同时利用 pdfminer.six 保留文档排版结构，辅助判断内容逻辑关系；

（3）对于 excel 文档，使用 pandas 读取表格数据，提取表头、数据内容及单元格注释，将每类数据主题（如 “2025 年 1 月理财产品销售数据”）对应的表格内容转化为结构化文本，标注数据维度与单位；

（4）对于 ppt 文档，通过 python-pptx 提取每页的文本、图片与图表信息，按

“幻灯片标题 - 内容模块” 拆分，图表需提取数据与图表类型（如柱状图、折线图），并转化为文字描述（如 “2024 年 Q4 信贷投放金额：个人贷款 50 亿元，企业贷款 80 亿元”）；

（5）对于 png、jpeg 图像文档，先使用 OCR 工具提取文字内容（如图表中的数据标签、流程说明文字），再结合图像识别模型（如 ResNet、YOLO）判断图像类型（图表、流程图、宣传图），针对图表提取数据关系，针对流程图提取步骤与逻辑节点，转化为结构化知识。

2. 知识提取与结构化

（1）采用 “关键词提取 + 实体识别 + 关系构建” 的方式处理解析后的文本内容：使用 TF-IDF、TextRank 等算法提取每个文档的核心关键词（如 “个人住房贷款”“年利率”“还款方式”）；

（2）利用金融领域预训练模型（如 FinBERT、BERT - 金融版）进行实体识别，标注出文档中的金融实体（如产品名称、金额、政策编号、时间）；

（3）基于依存句法分析与语义角色标注，构建实体间的关系（如 “个人住房贷款 - 年利率 - 4.5%”“贷款申请 - 所需材料 - 身份证”），将非结构化文本转化为 “实体 - 关系 - 实体” 的三元组，或 “问题类型 - 答案内容” 的键值对，形成结构化知识库，同时记录每个知识点的来源文档路径与页码，便于溯源。

（二）用户问题意图理解环节

针对多类型用户问题，需精准识别问题意图与核心需求，具体步骤如下：

1. 问题预处理

（1）对用户输入问题进行文本清洗，去除特殊符号、语气词（如 “呢”“呀”），统一大小写与数字格式；

（2）采用分词工具（如 jieba、THULAC）对问题分词，结合金融领域词典（如《金融术语大全》）优化分词结果，避免 “信贷” 被拆分为 “信”“贷” 等错误情况。

2. 问题类型分类

（1）构建分类模型（如基于 BERT 的文本分类模型），将问题分为多意图、推理、多跳、细节、长文本、总结六大类。训练时需结合竞赛提供的标注数据，标注每个问题的类型标签，例如：

1）多意图问题：标注多个核心需求标签（如 “贷款利率 + 申请材料”）；

2）推理问题：标注推理所需的规则标签（如 “客户资质 + 额度计算”）；

3）多跳问题：标注检索步骤标签（如 “监管政策→产品规则→客户要求”）。

（2）对于长文本问题，需先进行文本摘要（如采用 TextRank 或 PEGASUS 模型），提取核心疑问点，再进行类型分类，避免冗余信息干扰意图判断。

3. 核心需求抽取

（1）利用命名实体识别模型提取问题中的关键实体（如 “个人住房贷款”“2025 年”“50 万元”）；

（2）结合语义解析技术（如依存句法分析、语义角色标注），确定问题的核心动作与需求（如 “查询 - 利率”“判断 - 合规性”“总结 - 增长原因”），形成“关键实体 + 核心需求”的意图表示，为后续检索提供方向。

（三）知识点检索与排序环节

根据问题意图，从结构化知识库中检索相关知识点，并按相关性排序，输出前 3 个符合要求的知识点，具体步骤如下：

1. 知识库索引构建

（1）采用倒排索引技术，将结构化知识库中的知识点按关键词、实体、问题类型构建索引，例如：为“个人住房贷款年利率 4.5%”知识点，构建“个人住房贷款”“年利率”“4.5%”等关键词的倒排索引，记录知识点 ID 与来源；

（2）对于推理类与多跳类问题，构建知识图谱索引，将实体间的关系（如“贷款额度-依赖-客户收入”“监管政策-约束-产品设计”）存储为图结构，便于多步关联检索。

2. 多策略检索

（1）细节类问题：基于关键词匹配与实体匹配进行精确检索，例如问题 “某理财产品起购金额”，检索包含 “理财产品名称 + 起购金额” 实体的知识点；

（2）多意图问题：按拆分的多个意图分别检索，每个意图获取 Top2 相关知识点，再进行去重与合并，确保覆盖所有意图；

（3）多跳问题：采用 “逐步检索 + 关联扩展” 策略，例如问题 “监管政策对理财产品风险评级的要求，我行对应产品有哪些”，第一步检索 “监管政策 - 理财产品 - 风险评级” 知识点，第二步根据第一步结果中的风险评级标准，检索“我行产品 - 风险评级 - 符合标准” 的知识点；

（4）推理类问题：先检索推理所需的规则知识点（如 “贷款额度 = 月收入 ×12×50%”），再结合问题中的实体数据（如 “客户月收入 1 万元”）进行逻辑计算，生成推理结果知识点；

（5）总结类问题：检索与问题主题相关的多个知识点，采用摘要模型（如 BART、T5）进行信息融合与归纳，生成简洁的总结知识点；

（6）长文本问题：基于摘要后的核心需求检索，同时结合长文本中的上下文信息（如客户历史业务记录），筛选关联性更强的知识点。

3. 知识点排序与筛选

（1）构建多维度排序模型，综合考虑 “关键词匹配度”“实体重合度”“语义

相似度”“来源文档权威性”（如监管政策文档权重高于普通产品手册）等因素，采用线性加权或机器学习模型（如 XGBoost、LightGBM）计算知识点与问题的相关性得分；

（2）对检索出的知识点按相关性得分降序排列，筛选前 3 个知识点，同时检查每个知识点字数是否≤1500 字，若超出则进行精简（保留核心信息，去除冗余描述），确保符合输出要求。

十一、参考资源

为帮助参赛团队高效开展研发工作，提供以下参考资源，涵盖工具、模型、数据集与文献，参赛团队可根据需求选择使用：

（一）文档解析工具

1. 文本类文档工具：python-docx（word 解析）、pdfplumber（pdf 文本提取）、pandas（excel 解析）、python-pptx（ppt 解析）、markdown-it-py（markdown 解析），均为开源工具，支持 Python 语言，文档完善且社区活跃，可通过 PyPI 直接安装。

2. 图像类文档工具：Tesseract-OCR（开源 OCR 工具，支持多语言文字提取，可结合 pytesseract 库在 Python 中调用）、百度智能云 OCR（提供金融场景定制化 OCR 服务，支持表格、手写体识别，有免费调用额度）、YOLOv8（开源图像识别模型，可用于图像类型分类与图表元素检测）。

（二）预训练模型与框架

1. 金融领域预训练模型：

（1）FinBERT（基于 BERT 优化的金融领域模型，支持金融文本分类、实体识别，可在 Hugging Face 平台下载）；

（2）金融 BERT-wwm（中文金融领域预训练模型，由哈工大讯飞联合实验室发布，适合中文金融文本处理）；

（3）ALBERT - 金融版（轻量级预训练模型，训练速度快，适合资源有限的参赛团队）。

2. 深度学习框架：PyTorch、TensorFlow，均支持预训练模型的微调与部署，提供丰富的工具库（如 TorchText、TensorFlow Hub），便于构建意图理解与检索模型。

（三）技术文献与教程

1. 核心技术文献：

《基于 BERT 的金融文本意图识别研究》（探讨预训练模型在金融问题意图理解中的应用）；

《多跳知识图谱检索在金融问答系统中的实践》（介绍多跳检索技术的实现思路与优化方法）；

《多格式文档结构化解析技术综述》（总结各类文档解析的关键技术与难点解决方案）。

2. 实操教程：

Hugging Face 官方教程（包含预训练模型微调、文本分类、问答系统构建等实操指南）；

Python 金融文本处理实战教程（涵盖文档解析、实体识别、关键词提取等代码案例）；

知识图谱构建与检索教程（讲解基于 Neo4j 的知识图谱存储与多跳检索实现）。

（四）工具平台推荐

1. 模型训练与部署平台：Google Colab（免费 GPU 资源，适合小规模模型训练）、阿里云 PAI（提供大规模计算资源与模型部署工具，支持竞赛模型快速上线测试）；

2. 知识库管理工具：Elasticsearch（开源搜索引擎，支持高效的关键词检索与索引构建，适合结构化知识库存储）、Neo4j（图形数据库，适合知识图谱存储与多跳检索）；

3. 效果评估工具：NLTK（提供文本相似度计算、准确率评估等功能）、Scikit-learn（支持分类模型的精度、召回率计算，可用于意图识别模型评估）。

十二、提交产物要求

（一）初赛阶段

1. 提交内容：仅需提交 result.json 答案文件

2. 要求：

（1）参赛者在测试集发布后的一定时间窗口内（如 48 小时），通过竞赛平台提交一个 JSON 格式的答案文件。

（2）该文件必须包含对测试集中所有问题的回答，每个问题返回 3 个知识点。文件格式必须严格遵循赛题说明中的规定。

（3）此阶段仅需提交答案文件，无需提交模型、代码或报告。

3. 目的：筛选效果优异的队伍进入决赛。

（二）决赛阶段（针对初赛排名靠前的队伍）

1. 提交内容： 完整作品包，包括：

（1）技术方案报告（PDF 格式）

（2）全部源代码（包含依赖环境说明）

（3）训练/微调后的模型权重（或提供下载脚本）

（4）Docker 镜像文件（用于运行整个系统）

2. 要求：

（1）所有材料必须在规定时间内（如初赛结果公布后 72 小时）提交至指定位置。

（2）技术方案报告需详细、真实地阐述算法原理与实现细节：

1）模型参数格式：需统一采用以下格式之一，且需在技术方案报告中明确标注：

①PyTorch 框架：模型权重文件为.pth 或.pt 格式，需包含完整的模型结构配置文件（如 model_config.json），确保加载后可直接用于推理；

②TensorFlow 框架：模型权重文件为.h5 格式或 SavedModel 格式（包含 assets、variables、saved_model.pb 等文件），需提供模型输入输出的维度说明及数据类型定义。

模型参数文件需压缩为 ZIP 格式，单个压缩包大小不超过 20GB，且需附带 MD5 校验值供完整性验证。

3. 源代码必须可读、可运行，并提供清晰的说明文档：需包含 requirements.txt 文件，明确列出所有依赖库的名称及版本（如 torch==2.1.0、pdfplumber==0.10.3）；代码目录结构需清晰，包含 README.md 文件，详细说明各模块功能（如文档解析模块、意图理解模块、检索排序模块）、运行入口及参数配置方式。

4. 主办方将使用以下统一软硬件环境对提交的 Docker 镜像及模型进行效果复现与性能测试，具体信息如下：

（1）硬件环境：

• CPU：Intel Xeon Gold 6348 @ 2.60GHz（2 颗，共 40 核 80 线程）；

• 显卡：NVIDIA A100（80GB 显存，2 颗，支持 CUDA 12.1）；

• 内存：256GB DDR4 3200MHz；

• 存储：2TB SSD（用于存放数据集、模型文件及运行日志）。

（2）软件环境：

• 操作系统：Ubuntu 22.04 LTS；

• Docker 版本：24.0.7；

• CUDA 版本：12.1；

• CuDNN 版本：8.9.2；

• Python 版本：3.10.12；

• 基础依赖库：PyTorch 2.1.0、TensorFlow 2.15.0、pandas 2.1.4、numpy 1.26.3（其他依赖库以参赛队提交的 requirements.txt 为准，主办方将按文件内容安装）。

（3）验证流程：主办方将在上述环境中加载参赛队提交的 Docker 镜像，使用测试集（500 个用户问题）执行推理，分别计算 Top3 检索正确率与查询响应延迟，与参赛队提交的初赛答案及技术方案报告中的性能数据进行一致性验证。

• 目的：全面验证方案的真实性、可复现性和创新性。

十三、奖金设置

为了鼓励参赛选手参赛积极性，本赛题根据总决赛成绩，对成绩排名前三名的参赛团队设置奖金。

1. **冠军奖：**第 1 名，奖金 20000 元/每团队
2. **亚军奖：**第 2 名，奖金 10000 元/每团队
3. **季军奖：**第 3 名，奖金 5000 元/每团队