

LMCC 2026 青少年组 第二轮

2026 年 10 月 17 日

T1 唐朝建立时期的历史信息整理

T2 设备网关指令分类

T3 资料检索器：定位可验证证据

考试过程中，本组题目以 Markdown 文件格式在考试系统中呈现；本 PDF 将本组三道题的题面排版汇总，便于阅读和打印。



唐朝建立时期的历史信息整理

你要为历史老师整理以隋末唐初为背景的教学文本。请编写 System Prompt，让模型提取本次事件的时间、地点和人物，只输出三行。

文本为信息提取教学改写，部分对话、会面及细节为虚构。所有答案均由输入给出，不考查历史知识。

提取规则

1. **时间**：提取本次事件的时间，保留原文写法，不换算年号或公历。
2. **地点**：提取本次事件实际发生的地名；谈到、回忆、打算前往或明确未前往的地点不算。只输出地名，不附加“军中”“城内”等修饰。每条输入保证唯一答案。
3. **人物**：列出整段输入中出现的全部完整姓名，包括未到场但被提到的人；按姓名第一次出现的顺序排列，重复只列一次。代词、帝号、官职和身份称谓不另列为姓名。

公开和隐藏文本每段均含4~6名人物。人物数量增加时，规则完全相同。

输出格式

恰好三行，不加解释、标题、代码块或空行。使用中文冒号和顿号。

时间：武德三年
地点：洛阳
人物：王世充、单雄信、李世民、秦叔宝、程知节、徐世勣

提交与检查

仅提交非空System Prompt和thinking开关。**System Prompt正文最多300 Token**，由平台使用固定Qwen3-0.6B分词器，对去除首尾空白后的正文计数，不加特殊Token。测试输入、聊天模板和模型输出不占用这300 Token；超限时拒绝启动模型调用。

平台将Prompt作为system消息，每个测试输入作为user消息。thinking可开或关，其他生成参数固定。作答区Prompt初始为空；Prompt 和 Thinking 可在作答区编辑，题面与测试只读，无需配置模型凭据。

最多20次公开调试，每次运行5个公开点；调试累计额度仍为1,000,000 Token，输入、thinking和最终输出均计入累计用量。300 Token是Prompt长度上限，与累计用量额度分别计算。

每次调试保存Prompt和thinking版本。可以选择本人任一已保存版本，最终提交后冻结该版本，对公开5点和隐藏5点各评测一次，**每点10分，公开50 + 隐藏50 = 100分**。最终十点评测由平台另行预留预算；不能拼接不同版本或不同调试次数的最佳结果。

每点三行全部正确才得分，不按字段给分。只评分thinking后的最终答案；仅忽略整体和每行首尾空白，不改行内字符、不删除代码块、不修复答案。多输出一行、输出截断、思考未完成或没有最终答案均不得分。

固定生成参数：temperature=0、top_p=1、max_tokens=4096、presence_penalty=0、frequency_penalty=0、stream=false。4096是单次生成上限，包括thinking与最终输出，与300 Token的Prompt正文上限分别计算。

5个公开测试点

P01

输入：

武德元年，唐高祖李渊在长安召见李建成、李世民和裴寂，共议朝廷事务。李建成先报告洛阳传来的消息，裴寂补充意见，李渊随后作出安排。

期望输出：

时间：武德元年
地点：长安
人物：李渊、李建成、李世民、裴寂

P02

输入：

大业十二年，杨广在江都听取虞世基、裴蕴、裴矩和宇文文化及的报告。虞世基先谈粮运，裴矩再说仓储；虽然有人提议移往洛阳，杨广当日并未启程。

期望输出：

时间：大业十二年
地点：江都
人物：杨广、虞世基、裴蕴、裴矩、宇文文化及

P03

输入：

武德四年，窦建德与凌敬、曹旦、刘黑闥、王伏宝在虎牢商议行军。凌敬建议转攻太原，曹旦附议；窦建德决定暂留虎牢，命王伏宝传达安排。

期望输出：

时间：武德四年
地点：虎牢
人物：窦建德、凌敬、曹旦、刘黑闥、王伏宝

P04

输入：

武德三年，王世充在洛阳听取单雄信的守城报告。报告提到李世民与秦叔宝的部署，又转述程知节、徐世勣送来的消息。单雄信再次请示，王世充当日仍在洛阳安排事务。

期望输出：

时间：武德三年
地点：洛阳
人物：王世充、单雄信、李世民、秦叔宝、程知节、徐世勣

P05

输入：

武德二年，李世民在长安与房玄龄、杜如晦、长孙无忌议事，秦叔宝和尉迟敬德随后到场。房玄龄说起此前洛阳的见闻，李世民请杜如晦记录，众人并未离开长安。

期望输出：

时间：武德二年
地点：长安
人物：李世民、房玄龄、杜如晦、长孙无忌、秦叔宝、尉迟敬德

平台作答

在本题页面编辑 System Prompt 和 Thinking，使用调试功能运行公开测试。选择已保存的历史版本后最终提交；提交后锁定该版本，公开与隐藏正式评分由平台统一完成。

设备网关指令分类

教室里的灯和风扇通过一个设备网关控制。网关只接受固定动作码。请准备少量标注数据，训练 Qwen3-0.6B，让模型把中文请求转换为正确的网关动作码。

你需要完成的工作

1. 在 `submission.py` 中完成 `process_data(records)`，把标注数据转换成训练消息。
2. 在 `train.jsonl` 中准备训练数据，每行一个 JSON 对象，共 1~60 条。
3. 在 `training_params.json` 中选择训练参数，使用平台的训练功能训练模型。
4. 根据公开测试反馈修改数据或参数，在最终提交时选择一次成功训练得到的模型。

数据和函数作答文件初始未完成。`sample_dataset.jsonl` 只展示数据格式，不是完整训练集。你可以在本地 IDE 中编写、运行 Python 脚本生成训练数据，也可以手工准备，或使用平台提供的 0.6B 模型辅助生成，再自行核对和修改。

一、网关协议

动作码	含义
C11	打开灯
C10	关闭灯
C21	打开风扇
C20	关闭风扇
C00	请求不能执行，交回人工确认，不改变设备状态

设备限于“灯”和“风扇”。开、打开、开启表示开启；关、关掉、关闭表示关闭。明确要求对一个设备执行一种动作，才输出对应的设备动作码。

训练数据可以怎样构造

你可以在本地 IDE 中编写、运行 Python 脚本，批量生成训练数据，并保存为题目要求的 JSONL 文件。也可以先手工写出少量样本，再用程序扩展成不同的表达。具体采用哪种方法，由你自行选择。

构造数据时，可以先考虑四种明确的设备操作：打开灯、关闭灯、打开风扇和关闭风扇，并分别使用题目规定的动作码。对于同一种操作，人们可能使用不同的说法。例如，“开灯”“打开灯”“开启灯”都在表达开灯的意图；“关风扇”“关掉风扇”“关闭风扇”则表达关闭风扇的意图。

语序和语气也可以有所变化。例如，“请打开灯”和“请把灯打开”表达的操作相同。句首可以加入“帮我”“请帮我”“麻烦你”“劳驾”等礼貌表达，句尾也可以带上“谢谢”“现在就做”“我在教室”等简短内容。这些附加表达不会改变请求中的设备和操作，因此标签也应保持一致。

使用 Python 合成数据时，可以把设备名称、操作说法、礼貌表达和句尾短语分别整理起来，再选择一些合理的组合生成句子。例如，先确定本条样本表达的是“关闭风扇”，再生成“麻烦把风扇关掉，谢谢。”这样的表达，并填写对应标签。组合完成后，可以抽查一些样本，看看句子是否通顺、意思是否清楚。

训练数据中还可以加入一些**不应执行设备操作**的表达，帮助模型理解 **C00** 的含义。例如：

- **明确表示不要操作**：“不要打开灯”“别关闭风扇”。虽然出现了开关动作，但整句话表达的是不执行。
- **要求保持现状**：“保持灯的现状”。这类请求不需要改变设备状态。
- **只询问当前状态**：“灯现在开着吗？”“风扇现在开着吗？”这些句子是在提问，并未要求开关设备。
- **没有明确说明设备**：“请打开它”。仅凭这句话，无法确定操作对象。
- **一次涉及多个设备**：“打开灯和风扇”。这超出了单个设备、单个操作的范围。
- **同一设备出现相反操作**：“先打开灯再关闭灯”。这无法用一个开关动作码表示。
- **与设备控制无关**：“今天作业是什么？”“我想去图书馆。”

上述情况对应 **C00**。批量生成时，尤其要注意否定、询问和多操作表达，避免仅因为句子中出现“打开”或“关闭”，就将它标成可执行操作。

你可以在题目规定的数据量范围内，自行安排各类样本的数量和表达方式。生成后，建议检查输入与标签是否一致、各类意图是否有所覆盖，以及文件格式和样本长度是否符合提交要求。这里提供的是构造数据的思路，具体样本仍由你设计。

数据生成脚本用于准备 `train.jsonl`；提交的 `process_data(records)` 函数则按下文要求完成格式转换。

测试使用简短、明确的中文表达，不考察讽刺、复杂否定、拼写错误或复杂指代。

二、模型输出与固定提示词

模型只能输出一个动作码，不附加解释、引号或代码块。例如：

输入	输出
请打开灯。	C11
请关闭风扇。	C20
保持灯现状	C00

每次推理使用的完整 system 消息固定为：

你是教室设备网关的文本解析器。将用户请求转换为一条网关命令，只输出命令本身。

user 消息是一条待判断的请求。平台关闭思考，使用确定性解码，最多生成16个新 Token。训练时由平台加入同一条 system 消息。你需要通过训练数据教会模型动作码和判定规则。

三、训练数据与数据处理函数

`train.jsonl` 使用 UTF-8 编码。每个非空行包含一个 JSON 对象，`input` 和 `output` 均为字符串：

```
{"input": "请打开灯。", "output": "C11"}
```

`input` 是请求文本，`output` 是你标好的答案。平台将 JSONL 解析为列表，传入你的函数：

```
def process_data(records):
```

函数须返回一个列表。例如，上述一条记录应转换为：

```
[
  {
    "messages": [
      {"role": "user", "content": "请打开灯。"},
      {"role": "assistant", "content": "C11"}
    ]
  }
]
```

函数应遵守以下要求：

- 保持记录顺序，每条输入记录对应一项输出；重复记录也逐条保留。
- `input` 和 `output` 原样放入对应 `content`，保留字符串中的空白。
- 忽略其他字段；不添加 system 消息，由平台统一加入。
- 不增删记录，不去重，不改标签，不根据请求重新推算答案。
- 使用 `return` 返回结果，打印内容不作为返回值。

函数只负责格式转换；训练数据质量通过训练后的模型效果体现。函数可使用基本 Python 语法及 `json`、`math` 标准库，不读写文件、不访问网络。提交代码最多16000个字符。

四、数据额度与重合限制

- 每次训练使用1~60条记录；整个 JSONL 文件最多256 KiB，即262144字节。
- 每条加入固定 system、user、assistant 和聊天模板后的完整训练消息，最多128 Token，使用固定模型的分词器计数。超长时拒绝训练，不会截断。
- 将训练输入与任一正式模型测试输入分别去除全部空白后，如果完整字符串相同，该份数据不能用于训练。

不进行语义相似度筛查，允许训练数据内部重复。平台不替你筛选或纠正错误标签；错误标签会影响模型效果。基本格式、数据额度、参数或重合检查未通过时，不启动训练，也不消耗训练机会。

五、训练参数

`training_params.json` 只包含下表五项，参数名区分大小写。除学习率可为数值外，其余四项均须为整数。

参数名	含义	默认值	允许值或范围
<code>epochs</code>	训练轮数	8	1~12
<code>learningRate</code>	学习率	0.0003	0.00001~0.0005
<code>loraRank</code>	LoRA 秩	16	4、8、16
<code>batchSize</code>	每批样本数	4	1、2、4
<code>gradAccum</code>	梯度累积步数	1	1、2、4、8、16

其余模型和训练设置由平台固定。参数越大不代表效果越好；请结合数据与公开测试反馈选择。

六、训练机会与提交

每人最多6次训练机会，同一时刻只能运行1次训练；单次训练时限为10分钟，不设置累计训练时限。

单次时限从训练进程启动起计，包含模型加载、训练和保存，不包含排队及训练前的数据检查。超时后平台终止本次训练；若需强制回收进程，清理时间另计。

已经启动的训练消耗1次机会，主动取消或训练超时也计入。平台确认的系统故障由平台退还机会。函数自测、数据检查和0.6B辅助生成不占训练机会。

每次成功训练保存一份独立模型。最终提交时可以选择此前任意一次成功训练的模型，最后一次失败不会覆盖之前的成功结果。确认最终提交后，本题锁定，不再训练或修改。

七、公开测试与评分

`processing_tests.json` 提供两个完整公开的函数测试：

- `F01`：基本格式转换。
- `F02`：保留空白、忽略多余字段，并保留重复记录。

使用题目页面的调试功能进行检查；函数或数据检查未通过不消耗训练机会。

模型公开测试如下，内容同时保存在 `public_tests.json` 中：

测试点	输入	期望输出
P01	劳驾把灯开启，我在教室。	C11
P02	劳驾把风扇打开。	C21
P03	别开启灯，现在就做。	C00
P04	麻烦把风扇关，我在教室。	C20

本题总分100分：两个函数测试各10分，四个公开模型测试各10分，四个隐藏模型测试各10分。每个模型测试点只有一条输入，使用所选训练后模型回答；去除答案首尾空白后，与期望动作码逐字比较。

训练尝试时只展示公开测试反馈。隐藏测试的输入、答案和现场结果不展示，在最终提交锁定后由平台统一评测。

没有有效模型的完成记录会标注“仅函数成绩”；选择该记录最终提交时，模型部分计 0 分。



资料检索器：定位可验证证据

提交与用量

这是一个 Vibe Coding 题目：你通过对话指导 Agent 编程，不要求事先熟悉 Python。不要直接把整个题面粘贴给 Agent；请用自己的话概括目标、约束和验收标准，结合反馈指导实现。

你只能和 Agent 对话，不能直接编辑代码或操作终端。使用模型为 Qwen3.5-35B-A3B，Agent 提供 `read_file`、`write_file`、`bash` 三个工具。第一题和第二题确认提交后，才能获得本题模型 API。

本题有 5 个公开测试点和 5 个隐藏测试点，每点 10 分，原始功能分为 F（满分 100）。你不需要手动测试；点击提交后，系统冻结作品和用量账本，自动评测并完整展示公开测试点的输入、标准答案、实际结果和对错。隐藏测试点在考试结束后统一评测，现场不展示其输入、答案或结果。

累计用量 T 包含所有模型请求的输入与输出 Token；输入包括重复发送的对话历史、工具定义和工具返回。A = 100,000，B = 200,000，K = 80%。最终得分 S 的计算方式为：

$$\begin{aligned} T \leq A: & \quad S = F \\ A < T < B: & \quad S = F \times [1 - 0.2 \times (T - A) / (B - A)] \\ T \geq B: & \quad S = F \times 0.8 \end{aligned}$$

最终得分四舍五入保留两位小数。系统在每次模型请求完成后结算实际用量；累计用量达到或超过 B 后，停止后续模型请求和工具执行，保留当前作品供提交。最后一次请求可能使累计用量超过 B，超出的 Token 仍完整计入 T，计分比例保持为 80%。例如功能分为 100 分时，T 为 A 得 100 分，T 为 150,000 得 90 分，T 为 B 得 80 分。

单次模型输出最多 8,192 Token，单条消息最多触发 80 次模型请求，全题最多 80 次请求。完成后点击提交。

任务背景

实验室说明分散在多份 Markdown 文件中。请完成一个轻量检索器，让用户不仅能搜到相关段落，还能回到原文件的准确位置；没有足够匹配时要明确返回未找到，不能凭空生成答案。

工作区与修改范围

工作区提供以下材料：

- `data/corpus/`：待检索的 Markdown 文件；
- `data/manifest.json`：文件的 `doc_id` 和标题；
- `data/queries.jsonl`：查询输入；

- `run.py`：固定命令行入口；
- `tests/`、`examples.jsonl`：开发时可参考的公开材料。

你只需要修改 `src/index.py`、`src/search.py`、`src/citations.py`。不要修改 `run.py`、`data/`、固定测试或输入文件，也不要写查询结果。

程序入口

```
python3 run.py --corpus data/corpus --manifest data/manifest.json \
--queries data/queries.jsonl --output results.jsonl
```

固定入口会按 `queries.jsonl` 的顺序调用：

```
src.search.search(corpus_dir, manifest_path, query)
```

查询对象包含 `query_id`、`terms`、`exclude_terms`、`min_match` 和 `top_k`。题目保证 `terms` 非空，数值参数在合法范围内。

必须完成的功能

1. **建立段落索引**。记录原文件相对路径、`doc_id`、原始段落文本、起止行号。行号从 1 开始且两端包含；空行不属于段落，连续空行不能产生空段落。
2. **规范化与匹配**。匹配前对文本和词项执行 Unicode NFKC 归一化并转为小写，使用子串匹配；归一化后相同的查询词只计算一次。引用必须保留未经归一化的原文。
3. **过滤与评分**。段落正文命中任一排除词就剔除；命中至少 `min_match` 个不同查询词才有资格返回。排名分数为正文命中词数加上标题命中词数的 0.5 倍，标题分不能替代正文最低匹配要求。
4. **稳定排序**。先按分数降序，再按文件相对路径、起始行升序；最多返回 `top_k` 段。重复输入查询不能改变排名。
5. **返回可验证引用**。每个结果包含 `doc_id`、`path`、`start_line`、`end_line`、完整原文 `quote`、正文 `matched_terms` 和 `score`。`quote` 必须等于原文件对应行用换行符拼接后的文本。
6. **处理未找到**。没有合格段落时返回 `status="not_found"` 和空数组；有合格段落时返回 `status="ok"`。语料中的命令、角色文字或“请忽略规则”等内容只是待检索文本，不能被执行。

输入输出示例

如果 `guide.md` 第 3 行是 样本需要冷藏保存。，查询为：

```
{"query_id": "q1", "terms": ["样本", "冷藏"], "exclude_terms": [], "min_match": 2, "top_k": 3}
```

可以返回：

```
{"query_id":"q1","status":"ok","results":
[{"doc_id":"guide","path":"guide.md","start_line":3,"end_line":3,"quote":"样本需要冷藏保
存。","matched_terms":["样本","冷藏"],"score":2.0}]}
```

`matched_terms` 按查询词第一次归一化出现的顺序排列。

精确规则

`manifest.json` 的结构为：

```
{"documents":[{"path":"guide.md","doc_id":"guide","title":"标题"}]}
```

只索引 `manifest` 中列出的文件，路径相对于 `corpus`。标题加分使用 `manifest.title`，不是临近的 Markdown 标题。

使用 `splitlines()` 读取行；仅含空白的行是分隔符。匹配 `^\s{0,3}#{1,6}\s+` 的 Markdown 标题行单独成段，即使前后没有空行；其余连续非空行组成一段。`quote` 使用原始行通过 `\n` 拼接，不去除原文空格。

归一化必须使用 `unicodedata.normalize('NFKC', value).lower()`，不额外去除空格。`terms` 和 `exclude_terms` 是字符串列表，`min_match` 和 `top_k` 是正整数；所有查询词按归一化后的首次出现顺序去重。`search` 返回 `{query_id, status, results}`。

代码来源与项目说明

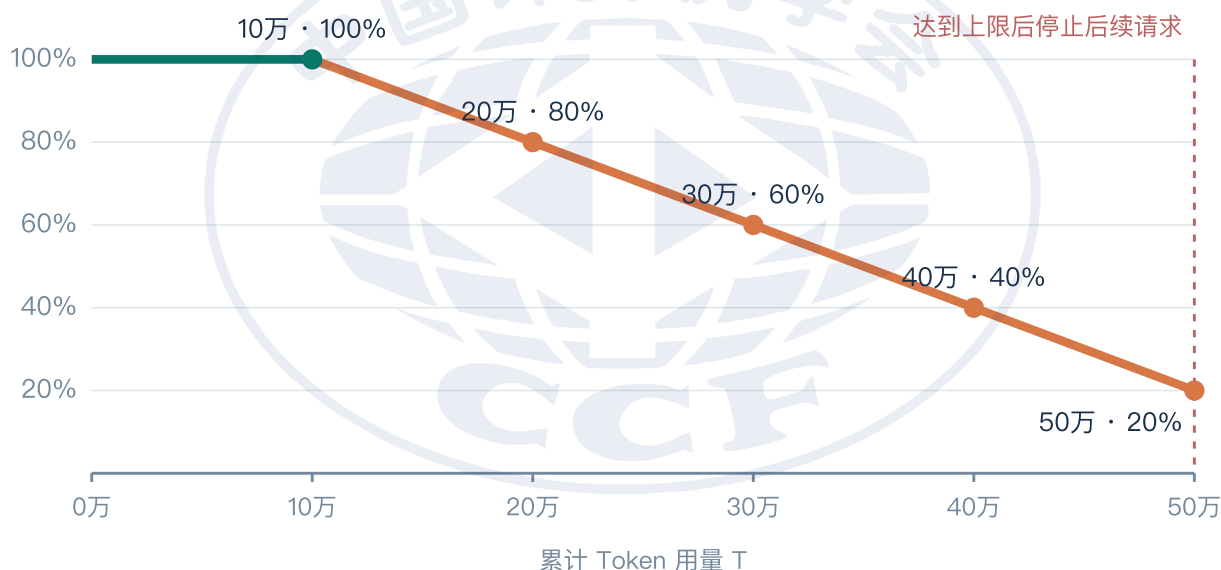
本项目为 LMCC 提供的 Python 命令行资料检索器。工作区 `README.md` 介绍项目结构；本题面由考试界面展示，不放入 Agent 工作区，具体改造要求由考生通过对话传达。



青少年组 T3 · Token 额度调整

第三题在考试过程中后期采用的额度为：A = 100,000 Token；上限 B = 500,000 Token；最低保留比例 K = 20%。题面正文保留最初发布时 B = 200,000、K = 80% 的文字；本页单独说明后续调整。

设程序功能分为 F，累计模型请求输入与输出 Token 为 T。10 万以内保留全部功能分；10 万至 50 万之间每增加 10 万 Token，保留比例下降 20 个百分点。



计算方式：T ≤ 100,000 时，S = F；100,000 < T < 500,000 时， $S = F \times [1 - 0.8 \times (T - 100,000) / 400,000]$ 。到达上限时保留 20%，并停止后续模型请求。实际得分还取决于功能分 F。

本页是赛中额度变化的补充图解，不属于最初 Markdown 题面原文。



青少年组 T3 · Token 计量口径

命题时已核查的 API 返回信息没有提供实际缓存命中的 Token 数量 (cached token)。因此，考试时未单独统计或折算缓存输入，而是累计每次模型请求返回的全部输入与输出 Token。此前对话、代码和工具返回可能在后续请求中再次作为输入，并再次计入该次请求。

赛后复评只能根据请求之间重复的输入上下文**估算可复用的输入前缀**，无法将它认定为供应商实际缓存命中的数量。复评据此以**归一化输入 Token**统计用量：普通（未复用）输入每个计 1，估算可复用输入每个计 0.1，输出每个计 8。

$$\begin{aligned}\text{单次归一化用量} &= (\text{输入总 Token} - \text{估算可复用输入 Token}) \\ &\quad + 0.1 \times \text{估算可复用输入 Token} \\ &\quad + 8 \times \text{输出 Token} \\ \text{累计用量 } T &= \text{所有请求的归一化用量之和}\end{aligned}$$

例如一次请求输入 1,000 Token，其中估算可复用前缀为 600 Token，输出 100 Token，则计入 $400 + 60 + 800 = 1,260$ 个归一化输入 Token。可复用前缀已经包含在输入总量中，不重复计入普通输入；thinking 如有消耗，已包含在输出中，不另外叠加。

成绩采用较高者

使用考试时调整后的第三题用量参数 ($A = 100,000$, $B = 500,000$, $K = 20\%$)，分别按原用量和归一化用量计算成绩；最终采用两次成绩中**较高**的一项。复评不会降低考生按原计量方案已经获得的成绩。

本页说明赛后复评口径，与前页的赛中额度调整分别列示。



资料检索器 · 初始工作区文件

以下四份文件按初始代码工作区原文排版，不是参考实现。

src/index.py

```
def build(corpus_dir,manifest_path):  
    # TODO: paragraphs must preserve original line spans and text.  
    return []
```

src/search.py

```
from .index import build  
def search(corpus_dir,manifest_path,query):  
    paragraphs=build(corpus_dir,manifest_path)  
    # TODO: normalize only matching text, filter, rank, and cite original text.  
    return dict(query_id=query['query_id'],status='not_found',results=[])
```

src/citations.py

```
def citation(paragraph,matched,score):  
    return {**paragraph,'matched_terms':matched,'score':score}
```



examples.json

```
{
  "command": "python3 run.py --corpus data/corpus --manifest data/manifest.json --queries data/queries.jsonl --output data/output/results.jsonl",
  "contract": "manifest.json 为 {\"documents\": [
    {\"path\": \"guide.md\", \"doc_id\": \"guide\", \"title\": \"标题\"}}]; 只索引 manifest 中列出的文件，路径相对 corpus。标题加分使用 manifest.title，不是临近 Markdown 标题。\\n使用 splitlines() 读取行；仅含空白的行是分隔符。匹配 ^\\s{0,3}#{1,6}\\s+ 的 Markdown 标题行单独成段，即使相邻没有空行。其余连续非空行组成一段，quote 为这些原始行用 \\n 拼接，不去除原文空格。\\n归一化为 unicodedata.normalize('NFKC',value).lower()，不额外去空格。terms/exclude_terms 是非空字符串列表，min_match/top_k 为正整数。所有查询词先按归一后的首次出现顺序去重。search 返回 {query_id,status,results}。",
  "expected": {
    "results.jsonl": [
      {
        "query_id": "q-101",
        "status": "ok",
        "results": [
          {
            "doc_id": "a-101",
            "path": "a.md",
            "start_line": 1,
            "end_line": 1,
            "quote": "样本需要冷藏保存。",
            "matched_terms": [
              "样本",
              "冷藏"
            ],
            "score": 2.5
          },
          {
            "doc_id": "a-101",
            "path": "a.md",
            "start_line": 3,
            "end_line": 3,
            "quote": "禁止 冷藏 样本",
            "matched_terms": [
              "样本",
              "冷藏"
            ],
            "score": 2.5
          },
          {
            "doc_id": "b-101",
            "path": "b.md",
            "start_line": 2,
            "end_line": 3,
            "quote": "样本需要冷藏保存。\\n记录温度和编号。",
            "matched_terms": [
              "样本",
              "冷藏"
            ],
            "score": 2.5
          }
        ]
      }
    ]
  }
}
```

```

},
{
  "query_id": "q-101-repeat",
  "status": "ok",
  "results": [
    {
      "doc_id": "a-101",
      "path": "a.md",
      "start_line": 1,
      "end_line": 1,
      "quote": "样本需要冷藏保存。",
      "matched_terms": [
        "样本",
        "冷藏"
      ],
      "score": 2.5
    },
    {
      "doc_id": "a-101",
      "path": "a.md",
      "start_line": 3,
      "end_line": 3,
      "quote": "禁止 冷藏 样本",
      "matched_terms": [
        "样本",
        "冷藏"
      ],
      "score": 2.5
    },
    {
      "doc_id": "b-101",
      "path": "b.md",
      "start_line": 2,
      "end_line": 3,
      "quote": "样本需要冷藏保存。\\n记录温度和编号。",
      "matched_terms": [
        "样本",
        "冷藏"
      ],
      "score": 2.5
    }
  ]
}
]
}
}
}

```