

存论文

1. 参考文档

1. 文献存储软件调研

- 目 [Zotero](#)

- 目 [Mendeley](#)

- 目 [EndNote](#)

- 目 [Citavi](#)

2. Zotero技术框架调研

- 目 [Zotero的技术架构和实现](#)

- 目 [ZoteroPDF内置阅读和标注功能的实现](#)

- 目 [Zotero插件技术架构和实现](#)

- 目 [Zotero浏览器插件的实现方式](#)

- 目 [Zotero网页抓取细节测试](#)

- 目 [Zotero注释和摘要和笔记联动原理](#)

- 目 [自定义URL协议及实现方法](#)

- 目 [支持格式](#)

- 目 [文献属性、标签、分类、类型（场景）](#)

- 目 [文献管理软件总结](#)

- 目 [借鉴和改进](#)

- 目 [管理思想](#)

- 目 [存储方式](#)

2. 现有文献软件 + 插件的 RAG 问题

2.1 三类 PDF 的现实差异

当文献系统以 **PDF 阅读器作为核心交互入口** 时，必须正视 PDF 在底层形态上的不一致性，这直接影响 AI 能力是否可用。

PDF 的三种现实形态

- 原生文本 PDF
- OCR 隐藏文本 PDF
- 纯图片 PDF

当前文献的提问仅针对原生文本PDF

2.2 回答质量问题

主要体现在三个层面。

1. Chunk 层级问题

- 实际使用的 chunk 多为：
 - 页级
 - 段落级
 - 固定 token 长度切分
- 这些 chunk 的本质是**“连续文本块”**，而不是科研中的：
 - 方法
 - 实验
 - 数据
 - 结论

结果是：

- AI 在回答时只能基于“哪段话看起来像答案”
- 无法区分该文本在研究结构中所处的位置与角色

2. Chunk 质量问题

- chunk 往往混杂多个语义单元：
 - 方法描述 + 实验条件 + 局部结论
- chunk 之间缺乏稳定边界：
 - 同一实验被切断
 - 条件与结果被拆散

直接后果：

- 向量召回结果不稳定
- 回答容易：
 - 断章取义
 - 条件错配
 - 结论泛化

3. 不支持知识图谱

- 由于 chunk 不是**对象级单元**：
 - 无法稳定抽取实体
 - 无法定义关系
- 最多只能得到：
 - “文本之间的相似性”
 - 而非“研究对象之间的关系”

结论是：

- 现有 chunk 体系 **只能做向量检索**
- **无法自然过渡到知识图谱或结构化推理**

因此，当前的 AI 问答本质仍是：**阅读辅助型 RAG，而非研究级 RAG**

3. 基于PDF阅读器的解决方案

依然采用PDF.js作为阅读器解决方案

3.1 正确的链路

代码块

- 1 PDF 阅读器 (UI 层)
- 2 - 渲染 PDF
- 3 - 高亮 / 框选 / 跳转
- 4 ↓ 产生
- 5
- 6 选区 (page + bbox)
- 7 - 纯空间信息，不依赖 PDF 是否可选中文本
- 8 ↓
- 9
- 10 证据对齐层 (必须自建)

```
11 - 将 bbox 映射为 evidence_block
12 - evidence_block = {page, bbox, text, confidence}
13   ↓
14
15 知识层 (AI 使用的真实对象层)
16 - Section
17 - Method
18 - Experiment
19 - Dataset
20 - Metric
21 - Result
22 - Limitation
23   ↓ 通过结构化抽取
24
25 对象级 Chunk (研究单元)
26 - 每个 chunk 对应明确研究对象
27 - chunk ↔ evidence_block 一一或多对一绑定
28   ↓
29
30 索引与推理层
31 - 向量索引 (语义检索)
32 - 知识图谱 (关系推理)
33 - 规则索引 (确定性约束)
34   ↓
35
36 回答生成
37 - 基于对象级 chunk / KG / 规则
38   ↓
39
40 证据回链 (UI 层反向映射)
41 - 根据 evidence_block
42 - 在 PDF 原页高亮对应 bbox
43
```

解决了什么关键问题:

- PDF 只负责“显示与坐标”，不再承担语义职责
- AI 的输入单位不再是 PDF chunk，而是研究对象
- 三类 PDF（文本 / OCR / 图片）在 UI 与 AI 层统一

3.2 三种PDF的解决方案

a. 原生文本 PDF

- 有 text layer
- UI 选区 ↔ 文本 ↔ bbox 可逆

b. OCR 隐藏文本 PDF

- 有 text layer，但可能漂移 / 错序
- 可用，但不完全可靠

c. 纯图片 PDF

- 无 text layer
- 原生无法选中
- 必须 OCR

3.3 使用 PDF 阅读器，并不阻碍翻译收费能力

已明确可行的形态：

1. 选中即译（tooltip / side panel）
2. 块级 overlay 译文（不写回 PDF）
3. 双栏对照阅读（左 PDF / 右译文文档）

关键点：

- 不要试图修改 PDF 本体
- 翻译作为“叠加层 / 并排层”存在



核心问题：

1. 规模化文献怎么管理？
2. 文献之间如何关联？
3. 文献与思考怎么连接？
4. 怎么拆分文献中的知识到论文？

4. 规模化文献怎么管理？

EndNote（最正面解决这个问题）

- **管理策略**
 - Smart Groups（规则自动归类）
 - Group Set → Group 的严格层级
 - 批量操作友好（导入、查重、全文挂载）
- **关键点**
 - 不依赖人记忆
 - 依赖**规则 + 批处理**
- **适用阶段**
 - 系统性综述
 - 长期学科积累
 - 机构级文献库

👉 EndNote 的“管理本质 = 规模与规则”

Zotero / Mendeley / Citavi 的取向

- Zotero：靠 Collection + Tag，**人工仍是主力**
- Mendeley：靠 Folder + Filter，**规模一大就吃力**
- Citavi：不追求“文献很多”，而是“知识可控”

5. 文献之间如何关联

Zotero（唯一原生支持“显式文献关系”的）

- **管理手段**
 - 关联文献（Item ↔ Item）
 - 明确区分：
 - PDF = 附件
 - 文献关系 = 关系边
- **关键点**
 - 文献不是孤立节点
 - 关系是可维护的对象
- **适用场景**
 - 系列研究
 - 方法演进

- 证据链追踪

👉 这正是我说的“Zotero 管的是文献之间怎么连”。

Zotero / Mendeley / Citavi 的取向

- Mendeley：没有文献↔文献关系模型
- EndNote：关系是“隐含在分组或引用里”，不可单独管理
- Citavi：更关注“文献 → 知识”，而不是“文献 ↔ 文献”

6. 文献与思考怎么连接

Mendeley（对这个问题最直接）

- 管理设计
 - Library（文献仓库）
 - Notebook（跨文献思考容器）
- 关键点
 - 思考不再绑定单一文献
 - Notebook 可以聚合多篇证据
- 适用场景
 - 综述构思
 - 问题导向阅读
 - 早期研究探索

👉 这正是“文献与思考怎么分、怎么连”。

Zotero / EndNote / Citavi 的差异

- Zotero：笔记仍然以文献为中心
- EndNote：几乎不解决“思考管理”
- Citavi：直接跳过“思考”，进入“知识”

7. 怎么拆分文献中的知识到论文

Citavi（唯一正面解决这个问题的）

- 管理核心
 - Knowledge Items（知识卡片）
 - 明确区分：
 - 原文摘录

- 转述
- 评论
- 想法

- **结构能力**

- 写作大纲 = 管理结构
- 知识直接挂到章节

- **适用阶段**

- 博士论文
- 专著
- 长周期深度研究

👉 所以我说 “Citavi 管的是：如何把文献拆成知识，再装配成论文”。

Zotero / EndNote / Citavi 的差异

- Zotero：笔记仍是“文献附属物”
- Mendeley：Notebook 偏思考，不是写作装配
- EndNote：写作阶段才介入，前期不拆知识