



可进化的知识系统

从理论到实践

在线研讨会 | 2026年4月30日

执行摘要 Executive Summary

本次分享旨在全面介绍利用大模型技术重构知识工程全链条的最新实践。核心思想是“好东西都是总结出来的”，强调知识系统应从传统的、预先设计的瀑布式模式，转向由Agent驱动的、能够自我迭代和进化的新模式。

01. 可进化的知识图谱与Wiki构造

提出Agentic Ontology新范式，通过永续Agent实现知识的自我审查与进化，让知识结构随业务发展持续演进。

02. 文档解析的基石 (PPX引擎)

介绍开源的PPX解析引擎，精准解决从非结构化PDF/图片文档中提取高质量结构化数据的行业核心难题。

03. 可进化的知识抽取系统

展示多智能体协作技术，实现从海量无标注文档中自动生成知识提取程序，无需人工干预即可完成知识沉淀。

04. 可进化的业务分析系统 (快查)

推出“快查”工具，将业务规则自动转化为可执行的核查系统，帮助企业实现业务运营与分析成本的指数级下降。

核心价值：展示了一套完整的、可进化的知识工程方法论，标志着知识系统构建正从“手工打造”迈向“自动化生成与进化”的新阶段。

目录 CONTENTS

01

可进化的知识图谱
与Wiki构造

从瀑布式到Agentic
Ontology

02

文档解析的基石
——PPX引擎

高质量结构化数据的源头

03

可进化的
知识抽取系统

从文档到提取程序的自动
化

04

可进化的业务分析系统——快查

自动化构建核查系统的Harness

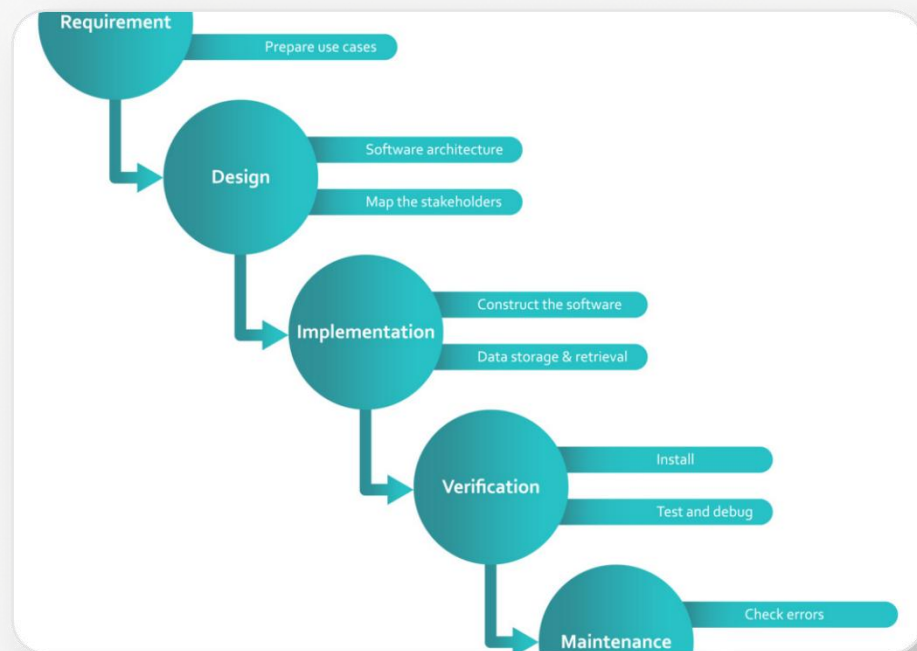
05

总结与展望

全链条解决方案与社区共建

01. 可进化的知识图谱与Wiki构造

传统知识工程的困境：瀑布式方法的终结



在大模型时代之前，知识工程领域长期被一种“瀑布式”的方法论主导。这种模式试图在项目初期就通过人工定义构建完整的知识体系，但在面对海量、快速变化的现实世界知识时，逐渐暴露出严重的局限性。



专家驱动

极度依赖领域专家
手工定义Schema
和复杂的逻辑规则。



预先设计

试图在项目启动初期
就设计好完整的
知识体系结构。



高昂成本

如CYC项目，单条
知识成本高达30美元，
耗时数十年。



不可扩展

知识数量受限于人力，难以覆盖
大规模常识。



僵化不变

Schema难以修改，无法适应
知识的动态演变。

新范式：Agentic Ontology —— 好东西是总结出来的

◆ 核心思想

Agentic Ontology: 知识库是生产出来的，不是设计出来的

核心定义

可进化的知识系统是一个能够从数据源永恒、不断地读取信息，并自主改变自身结构和内容的系统。其全生命周期——从数据到知识的升级过程及升级方法本身——均实现自动化。

On-the-fly

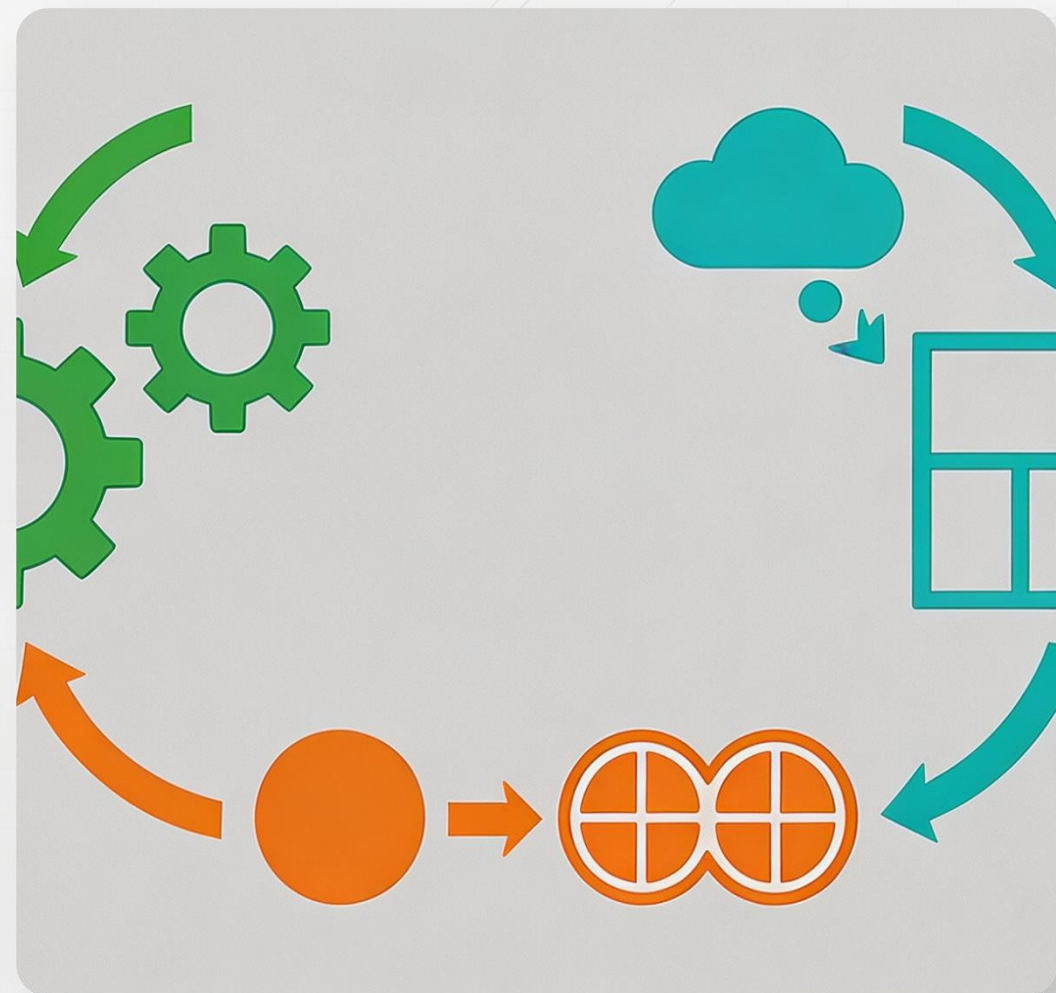
所有工作在动态执行中优化，如同“在飞行中更换引擎”。

Just-in-time

知识结构和规则非预先设计，而是根据数据处理的需要即时总结生成。

Bootstrap

每一步是下一步的基础，通过不断迭代从简单到复杂构建知识体系。



核心机制：永续Agent“管家 (Butler)”

实现知识系统进化的核心，是一个被称为“管家 (Butler)”的永续运行Agent系统，它像轻量级操作系统一样管理知识生态。

本质 Essence

一个有状态、有记忆的Agent队列系统。它类似于一个轻量级的操作系统，将计算机科学中的核心概念引入知识管理，包含：

- 进程调度 (Process)
- 任务队列 (Queue)
- 资源互斥 (Lock)
- 协作与竞争 (Competition)

核心功能 Functions

1. 自我审查 (Self-audit): 自动发现知识库错误，执行“内务整理 (Housekeeping)”动作进行修复。
2. 日志驱动反思: 维护详细运行日志，总结经验并持续优化执行策略。
3. 技能进化: 工作流沉淀为可复用、可进化的“技能(Skill)”模块，实现能力的自动扩展。

知识新形态 Ontology

从“名词”转向“动词”

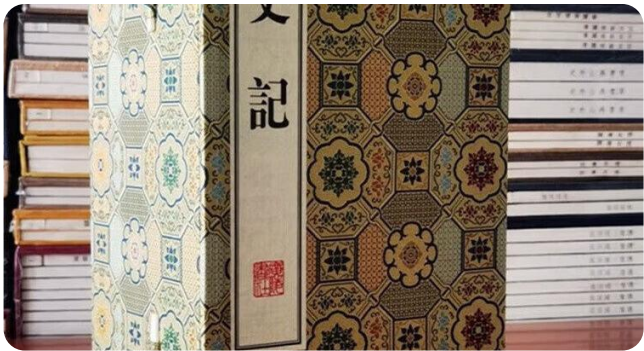
传统知识定义“是什么(事实)”，而 Skill 定义“怎么做(过程)”，成为系统的新本体。

三位一体知识体系：

事实 (Fact) + 叙事 (Narrative) + 技能 (Skill)

案例展示：三大开源知识库

我们将上述方法论应用于三个经典文本，构建了开源的Wiki知识库，所有前端页面也由机器辅助设计。



《史记》知识库

解析57万字，生成2万+页面，1.5万实体，17万三元组

时空关系挖掘

自动绘制历史人物战争轨迹地图、项羽分封十八路诸侯地理分布。

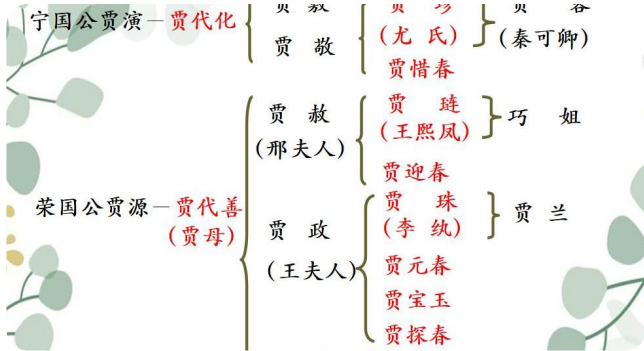


《三体》知识库

生成1000+页面，涵盖小说全部核心剧情与设定

叙事性知识挖掘

自动梳理危机纪元时间线、发掘小说前后呼应的伏笔与关键事件。



《红楼梦》知识库

生成2000+页面，内容随算法优化持续增长中

复杂关系挖掘

自动整理人物家族关系图谱、列出书中所有出现的食物并进行分类。

关键成果：成本的指数级下降

通过知识和技能的积累，知识库构建的效率大幅提升，成本显著下降，实现了知识的复利效应。从早期的“投石问路”到后期的“轻车熟路”，每一步积累都在为未来降本增效。

《史记》项目

耗时：约 2 个月

成本：约 **30,000 元**

—— 初期探索阶段 ——

《三体》项目

耗时：5 天

成本：约 **1,000 元**

—— 技术积累阶段 ——

《红楼梦》项目

耗时：1.5 天

成本：约 **100 元**

—— 成熟复用阶段 ——



结论：积累的知识越多，后续构建新系统的人工和资金投入就越少。
知识的复利效应已显现，可进化系统正在持续释放降本增效的价值。

02. 文档解析的基石——PPX引擎

为什么需要专门的文档解析引擎？



大模型的局限性



Token 消耗巨大

处理长文档成本高昂，大模型在上下文窗口内的效率随文本长度急剧下降。



幻觉与溯源难题

容易捏造信息，且无法精确定位内容在原文中的物理位置（Bbox），难以保证结果准确。



复杂布局上下文丢失

难以理解跨页、跨栏的复杂版式，易丢失段落间的逻辑关联。

企业级核心需求



高精度结构化提取

需准确提取表格、图片、公式、页眉页脚等非文本对象，并还原层级关系。



可溯源性要求

在金融、法律等严肃业务场景，必须能追溯回答的原文来源和具体坐标。



数据安全性与私有化部署

内部敏感资料无法出外网，要求支持本地私有化部署，确保数据不出域。

PPX引擎：面向知识工程的文档解析基础设施



PPX (皮皮虾) 是我们开源的、面向知识工程的 PDF 及图片结构化解析引擎，致力于解决复杂文档解析难题。



核心定位：知识工程上游基石

作为知识工程的上游基础设施，PPX 为 Retrieval-Augmented Generation (RAG)、知识抽取等下游任务提供高精度、高可用性的结构化数据支撑，打通非结构化数据到结构化知识的“最后一公里”。



技术架构：多模型协同解耦

摒弃单一视觉大模型依赖，采用多模型协同架构，有机结合OCR识别、智能版面分析、以及专项的表格/图表识别模型。模块化设计让系统更灵活，易于针对特定场景进行独立优化与扩展。



复杂对象解析

不仅能处理普通文本，更能精准解析各类复杂对象：包括无边框/多层嵌套表格、问卷表单、LaTeX数学公式，以及页眉页脚等特殊版式元素。



工程化部署友好

支持 CPU/GPU 混合部署模式，兼顾成本与性能；系统设计充分考虑 Agent 智能体的调用需求，支持水平扩展，易于在企业级生产环境落地。



可调试与可观测

全链路保留中间处理过程数据，为开发者提供清晰的调试依据。方便快速定位识别错误，持续迭代优化系统性能。



精确坐标溯源

解析结果附带精细的、字级别的 Bounding Box (Bbox) 坐标信息，支持在原始文档图片中进行精准定位与回溯，满足高要求的引用场景。

03. 可进化的知识抽取系统

业务目标：从文档到提取程序的自动化

系统核心在于：针对具有相似模式的文档（如民事裁定书、标准合同），自动总结通用信息提取逻辑，最终生成可直接运行的提取代码，实现流程自动化。



INPUT
输入

一批具有相似模式的
PDF/文档样例集合



OUTPUT
输出

提取关键信息的Python程序
(案号/当事人/判决结果等)

技术架构演进：从单一到协同

01 / 第一代：固定 Workflow

通过预设提示词模板，引导大模型直接生成代码。**缺点：**上下文与逻辑固定，面对复杂场景灵活性差。

02 / 第二代：单 Agent 系统

赋予 Agent 工具箱，支持其自主调查、验证与修改代码。**缺点：**在复杂逻辑下，Agent 容易发生“早停”，未完成任务就退出。

03 / 第三代：Agentic Workflow (当前架构)

引入多角色分工的多智能体协同机制，将复杂的抽取任务拆解，由不同专长的 Agent 配合完成，大幅提升准确率与稳定性。

核心工作流程：无标注启动，自动迭代



Supervisor (主管)

系统的“总指挥”，负责各 Agent 之间的整体调度与关键决策把控。



Business Agent (业务)

理解业务需求，自动定义 Schema 并完成无标注数据的初始标注。



Dev Agent (开发)

根据业务口径和标注数据，编写、测试和持续优化数据提取代码。



Evaluator (评估器)

运行自动化评估，提供准确率指标，判断代码是否达到上线标准。

01 定义与标注

Supervisor 调度 Business Agent，从无标注数据中自动生成 Schema 并完成标注。

02 初始评估

Supervisor 触发 Evaluator，对初始生成的数据进行基准准确率评估。

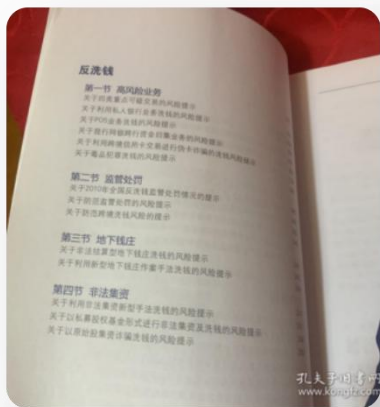
03 代码生成

Dev Agent 接收指令，基于业务口径和标注结果编写提取代码。

04 迭代闭环

若评估未达标，Evaluator 将反馈给 Dev Agent 进行修复，直到指标达标。

04. 可进化的业务分析系统——快查 (Knowledge Check)



业务背景：文档核查的挑战与价值

- 规则复杂**：源自法规与监管，逻辑严密且体系庞大。
- 文档多样**：财报、合同等篇幅长，结构非标准化。
- 价值巨大**：自动化可显著提升合规效率，规避操作风险。



核心洞察：知识建模

文档核查的自动化，本质上是将业务专家头脑中的隐性规则与逻辑，转化为机器可理解、可执行的标准化知识模型。



快查 (KC) 是什么？

它不是一个直接执行核查的终端应用，而是一个“自动化构建文档核查系统”的工具 (Harness)。



定位：工程师的替代者

快查工具不替代一线的合规核查人员，而是替代过去负责将专家知识翻译成代码的人类工程师与业务分析师，实现核查系统的快速生成与迭代。

工作流程：七大模块的自动化流水线



1. Bootstrap (初始化)

扫描工作空间，自动识别并解析输入的规则源文档和核查文档样本。



2. Rule Extraction

从自然语言规则源中，提取并解构出原子化、可执行的核查规则逻辑。



3. Skill Writing

自动将提取出的逻辑转化为标准化、可执行的 Rule Skills 代码片段。



4. Skill Testing

利用开发数据集对生成的 Skills 进行快速测试，自动识别错误并迭代优化。



5. Distillation (蒸馏)

将复杂模型能力“蒸馏”为轻量、高性能的专用 workflow，确保低延迟响应。



6. Full Test (全量测试)

在全量真实数据集上进行端到端验证，确保系统的准确性与稳定性。



7. Packaging (打包)

将所有模块产出打包为标准化容器镜像，提供 API 接口，实现一键部署。



传统模式：高昂的人力成本

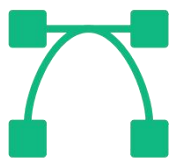
开发100条规则核查系统 \approx **1万元 / 规则** (需大量工程师参与)



快查模式：成本指数级下降

同等规模系统 \approx **500元 Token成本**，无需人工干预

哲学思考与未来愿景



Harness 设计哲学： “硬跟踪，软执行”

通过引擎强制追踪关键步骤和产出，确保 **Agent** 的工作不偏离轨道；但在具体执行方式上给予充分的自由度，鼓励其发挥创造性，发现更优的解决方案。



业务分析的本质

业务分析的过程，本质上是通过结构化拆解与专业知识建模，将一组旧文档（业务问题描述）转化为一组新文档（可执行的解决方案），实现信息的价值跃迁。



未来愿景：对标 Excel

Excel 是“可以变成任何业务工具的工具”。快查的目标是成为一个**可以自动化构建任何业务分析系统的工具**，让每一位业务专家都能将自己的方法论快速转化为可运行的自动化系统。

总结与展望

核心思想回顾：“好东西都是总结出来的”

无论是知识图谱、解析引擎、抽取系统还是业务分析工具，其核心都是构建一个能够自我学习、自我总结、自我进化的系统。



01 / 解析 (PPX)

将非结构化文档转化为高质量的结构化数据，奠定数据基石。



02 / 抽取 从结构化数据中自动提取知识模式并生成代码，实现知识挖掘。



03 / 构建 利用提取出的知识，构建可自我进化的知识图谱和Wiki系统。



04 / 应用 将沉淀的知识转化为可执行的业务分析系统，赋能业务价值。