HOW ONTOLOGIES ARE ACTUALLY BUILT · 上

一篇讲明白 ontology 到底怎么建出来

六步流程 · 每步的真实卡点 · 以及最容易缺的三样东西

心晴九天 · 小红书 howru999

小红书 · 关注更新

Agent 助理 · 扫码交流

系列第 3 期（上）· 承接第 1 期《本体论与企业AI》、第 2 期《数据中台差三层》

HOW ONTOLOGIES ARE ACTUALLY BUILT

一篇讲明白 ontology 到底怎么建出来

六步流程 · 每步的真实卡点 · 以及最容易缺的三样东西

前两期讲完，收到最多的一个问题

第一期讲本体论为什么是企业 AI 的地基，第二期讲数据中台离接上 Agent 还差三层。

两期讲完，最多的问题是：**道理我懂了，那这个本体，具体怎么建出来?**

这一期只回答这一个问题。六步流程不是我编的框架，是把 2023 年到现在这个方向上主要论文的做法，压到一条线上得到的。

> **本体构建这件事，难的不是把知识捞出来，难的是确认捞出来的东西没错。这是这两期的骨架。**

THE SIX STEPS

六步全景

点任意一步看它的位置。前五步在生产，第六步在确认。

1	2	3	4	5	6
定边界	抽实体	定类型	建层级	抽关系	验证
SCOPE	EXTRACT	TERM TYPING	TAXONOMY	NON-TAX RE	VALIDATE

前五步 — 把知识捞出来　　第六步 — 确认它没错

市面上绝大多数方案在第五步就交货了。而企业真正付钱的，是第六步之后的东西。

WHERE IT BREAKS

逐步拆：每一步卡在哪

点开每一步，看卡点、论文依据、自查项和不合格信号。

STEP 1 定边界

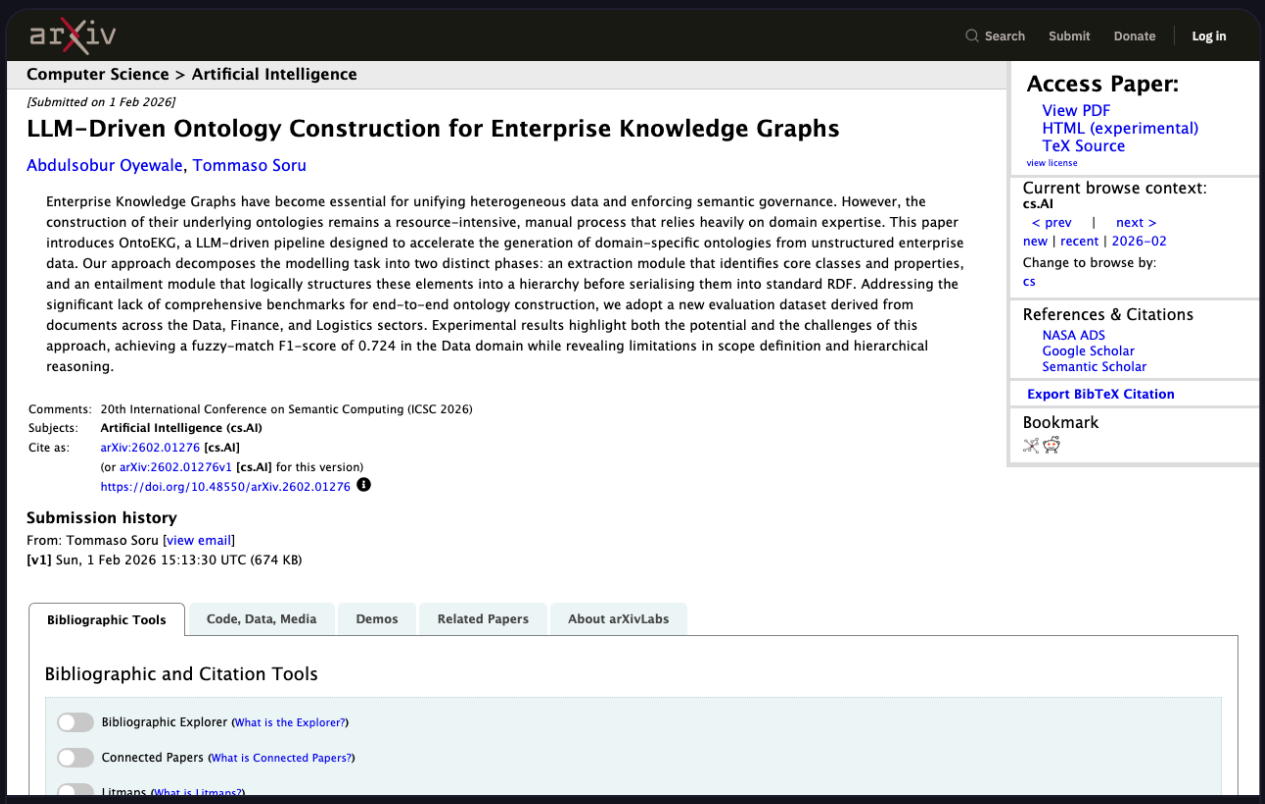
很多人以为第一步是画图，其实第一步是划范围。

卡点

大模型**没有能力自己判断这个本体该有多大**。你不明确告诉它边界在哪，它会一直往外扩 — 让它建一个采购本体，它能扩到供应链金融。这不是它贪心，是它不知道你的业务在哪结束。

论文依据

OntoEKG（arXiv:2602.01276，2026-02）列的第一条局限：LLM 无法自主确定合适的本体范围，需要人明确给出边界信号。



arXiv:2602.01276 摘要末句：revealing limitations in scope definition and hierarchical reasoning —— 范围界定与层级推理，正好对应第一步和第四步的卡点

自查

- [ ] 能用一句话说出范围吗（例如"只管进项抵扣，不管出口退税"）
- [ ] 这个范围是业务方定的，还是技术方猜的
- [ ] 范围外的东西，有明确的"不做"清单吗

> **不合格信号：**范围描述里出现"以及相关的""等等"。

结论

第一步不是技术活，是业务活。得有一个人说：我们这次只管进项抵扣。

STEP 2 抽实体

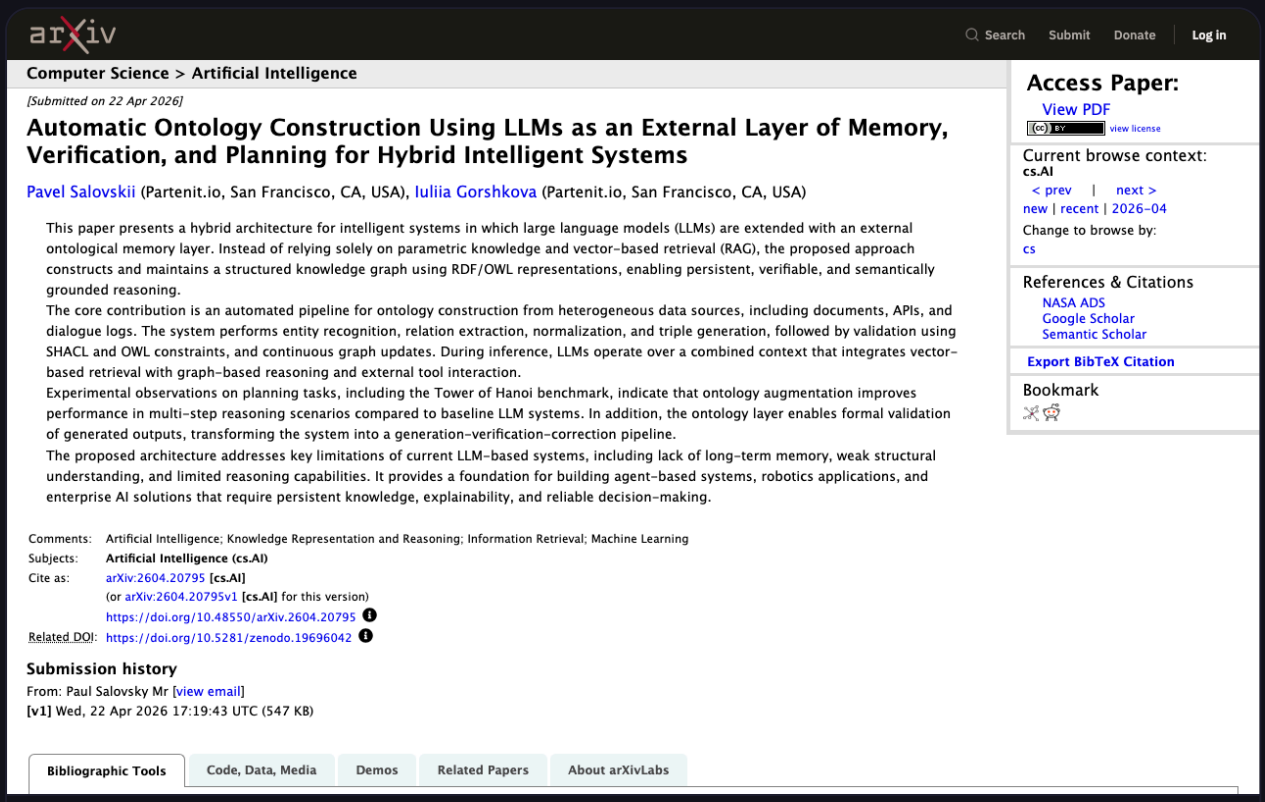
从文档里把名词找出来。发票、合同、纳税人、税额。

卡点

整个流程里**绝大多数错误都在这一步产生**。后面所有步骤都在放大它，而且越往后越难发现。

论文依据

Salovskii & Gorshkova（arXiv:2604.20795，Partenit.io，旧金山）在局限性里写：下游错误绝大部分源自实体识别阶段 — 类型给错、别名未归并、类型覆盖不全。



arXiv Search Submit Donate Log in

Computer Science > Artificial Intelligence

[Submitted on 22 Apr 2026]

Automatic Ontology Construction Using LLMs as an External Layer of Memory, Verification, and Planning for Hybrid Intelligent Systems

Pavel Salovskii (Partenit.io, San Francisco, CA, USA), Iuliia Gorshkova (Partenit.io, San Francisco, CA, USA)

This paper presents a hybrid architecture for intelligent systems in which large language models (LLMs) are extended with an external ontological memory layer. Instead of relying solely on parametric knowledge and vector-based retrieval (RAG), the proposed approach constructs and maintains a structured knowledge graph using RDF/OWL representations, enabling persistent, verifiable, and semantically grounded reasoning.

The core contribution is an automated pipeline for ontology construction from heterogeneous data sources, including documents, APIs, and dialogue logs. The system performs entity recognition, relation extraction, normalization, and triple generation, followed by validation using SHACL and OWL constraints, and continuous graph updates. During inference, LLMs operate over a combined context that integrates vector-based retrieval with graph-based reasoning and external tool interaction.

Experimental observations on planning tasks, including the Tower of Hanoi benchmark, indicate that ontology augmentation improves performance in multi-step reasoning scenarios compared to baseline LLM systems. In addition, the ontology layer enables formal validation of generated outputs, transforming the system into a generation–verification–correction pipeline.

The proposed architecture addresses key limitations of current LLM-based systems, including lack of long-term memory, weak structural understanding, and limited reasoning capabilities. It provides a foundation for building agent-based systems, robotics applications, and enterprise AI solutions that require persistent knowledge, explainability, and reliable decision-making.

Comments: Artificial Intelligence; Knowledge Representation and Reasoning; Information Retrieval; Machine Learning
Subjects: **Artificial Intelligence (cs.AI)**
Cite as: arXiv:2604.20795 **[cs.AI]**
(or arXiv:2604.20795v1 **[cs.AI]** for this version)
https://doi.org/10.48550/arXiv.2604.20795
Related DOI: https://doi.org/10.5281/zenodo.19696042

Submission history
From: Paul Salovsky Mr [view email]
[v1] Wed, 22 Apr 2026 17:19:43 UTC (547 KB)

Bibliographic Tools | Code, Data, Media | Demos | Related Papers | About arXivLabs

Access Paper:
View PDF
view license

Current browse context:
cs.AI
< prev | next >
new | recent | 2026-04
Change to browse by:
cs

References & Citations
NASA ADS
Google Scholar
Semantic Scholar

Export BibTeX Citation

Bookmark

arXiv:2604.20795 —— 用外部本体记忆层给大模型做长期记忆与验证，作者把 NER 阶段列为主要误差来源

一个真实的易混对

概念	判定依据
小规模纳税人	增值税上的身份，看年销售额是否超过 500 万元
小微企业	企业所得税和普惠政策上的概念，看从业人数、资产总额、应纳税所得额

两套完全不同的判定标准。一个企业可以是其中一个而不是另一个。第二步归错，后面全歪。

自查

- [ ] 同一个东西的不同叫法，有没有归成一个
- [ ] 业务方逐条看过这份清单吗
- [ ] 清单里有没有其实是数值、日期、状态的东西被误当成实体

> **不合格信号：**实体清单是模型跑出来直接用的，没人逐条看过。

STEP 3 定类型 term typing

判断你抽出来的这个词，它是个什么。

卡点：抽象层级

"小规模纳税人"是一个**类**（一群纳税人），不是某个具体企业。但大模型经常把它当成一个个体。

后果比听起来严重：在本体里，类可以有子类、可以往上挂规则；个体只能被规则约束。**你把类当成个体，这个节点下面就再也长不出东西，规则也挂不上去。**

论文依据

OntoEKG 第二条局限：模型有时会把本该是类的东西建成个体。

自查

- [ ] 逐条问："这个词指的是一群东西，还是一个东西"
- [ ] 是类的，能举出至少两个属于它的具体例子吗
- [ ] 是个体的，它属于哪个类，说得出来吗

> **不合格信号：**某个节点既没有子类也没有实例，且不被任何属性引用。

STEP 4 建层级

搞清楚谁是谁的子类。

卡点：方向

大模型会把父子关系搞反，而且它对"子类"的理解特别松 — 只要两个词经常一起出现，它就敢说这是子类。

危害是隐蔽的：本体是拿来推理的。你说 A 是 B 的子类，系统就认为凡是 B 的规则 A 都适用。**方向一反，规则就往错的方向传播 — 而且它不报错，只是给你错的答案。**

论文依据

OntoEKG 第三条局限：模型在层级推理时会搞错方向，并且使用了过于宽松的包含关系定义。

最好用的自查动作：念一遍

把每一对关系念成"**每一个【子】都是一种【父】**"。念得通就对。

✓ 每一个**小规模纳税人**都是一种**纳税人**
念得通 — 真的 is-a

✗ 每一张**发票**都是一种**发票台账**
念不通 — 这是部分与整体，不是父子

✗ 每一个**进项税额**都是一种**销项税额**
念不通 — 应纳税额 = 销项 − 进项，这是计算关系

> **不合格信号：**两个概念经常一起出现，就被判成了父子关系。

STEP 5 抽非分类关系

抵扣、视同销售、纳税义务发生时间 — 不是层级，是业务关系。

卡点：公认最难的一步

层级关系在语言里有痕迹 — "小规模纳税人"这个词里就带着"纳税人"三个字，模型能看出来。

但"一般纳税人可以抵扣进项税额"这件事，你在这两个词里找不到任何线索。

> **它不在词里，它在制度文件里。**

论文依据

*LLMs4OL Challenge @ ISWC（2025 年第二届）总结报告：LLM 在 term typing 和 taxonomy discovery 上表现不错，**非分类关系抽取始终是最差的一项**，因为它需要超越表面线索的语义理解。*

自查

- [ ] 每条关系，能指出它的依据在哪份文件的哪一条吗
- [ ] 关系两端的类型限制写了吗（谁可以抵扣，抵扣什么）
- [ ] 有没有关系是"看起来合理"但找不到依据的

> **不合格信号：**关系是靠常识推出来的，没有制度依据。

下期预告：我用实测数据推翻了这条"最难"的结论。同一个模型同一套题，**只靠通用知识准确率 60%，把制度原文附上去之后 100%**。差别不在模型，在有没有给材料。

STEP 6 **验证**

前五步在生产，这一步在确认。至少过三道：语法、逻辑一致性、结构缺陷。

卡点：最简单也最致命

绝大多数方案没有这一步。聚类那一派、两步走那一派、端到端微调那一派，大部分在第五步就交货了。

做得最完整那个方案的作者，自己写下的结论

"大模型不具备本体开发所需的程序性能力，缺少推理能力和领域专业知识，必须被嵌入到工作流工具里才能用。" — NeOn-GPT, ESWC 2024

自查

- ☐ 语法：文件能被标准工具解析吗
- ☐ 逻辑：有没有跑推理机，有没有自相矛盾的公理
- ☐ 结构：有没有孤立节点、缺少定义域值域的属性、多个概念共用一个标识
- ☐ 推理机报错时，能定位到具体是哪个概念出问题吗

> **不合格信号：**交付物里没有验证报告。

下期会讲这一步实测踩到的四个坑，包括一个反直觉的：**两个推理机会对同一个本体给出相反结论**，而只信一个的后果是模型会靠删规则来消除报错。

WHAT EVERYONE MISSES

六步之外，最容易缺的三样

我把主流开源方案都跑过一遍，缺的都是同样这三样。

⚠ 例外

规则能覆盖八成情况，**企业的钱全在剩下那两成里**。

增值税法第二十二条第（五）项：**购进并直接用于消费的**餐饮服务不得抵扣。 关键在"直接用于消费"五个字 — 买了转售给下游客户，是可以抵扣的。 这个限定没被建成显式例外，系统就会

⚙ 动作

只有实体、关系、度量的图，是一张**只读的图**。它能回答问题，不能改变业务。

企业买的是能改变世界的系统。要改变世界，动作必须是一等公民 — 而且每个动作都要能**授权**、能**留痕**、能**撤回**。 这三件缺

⚖ 验收标准

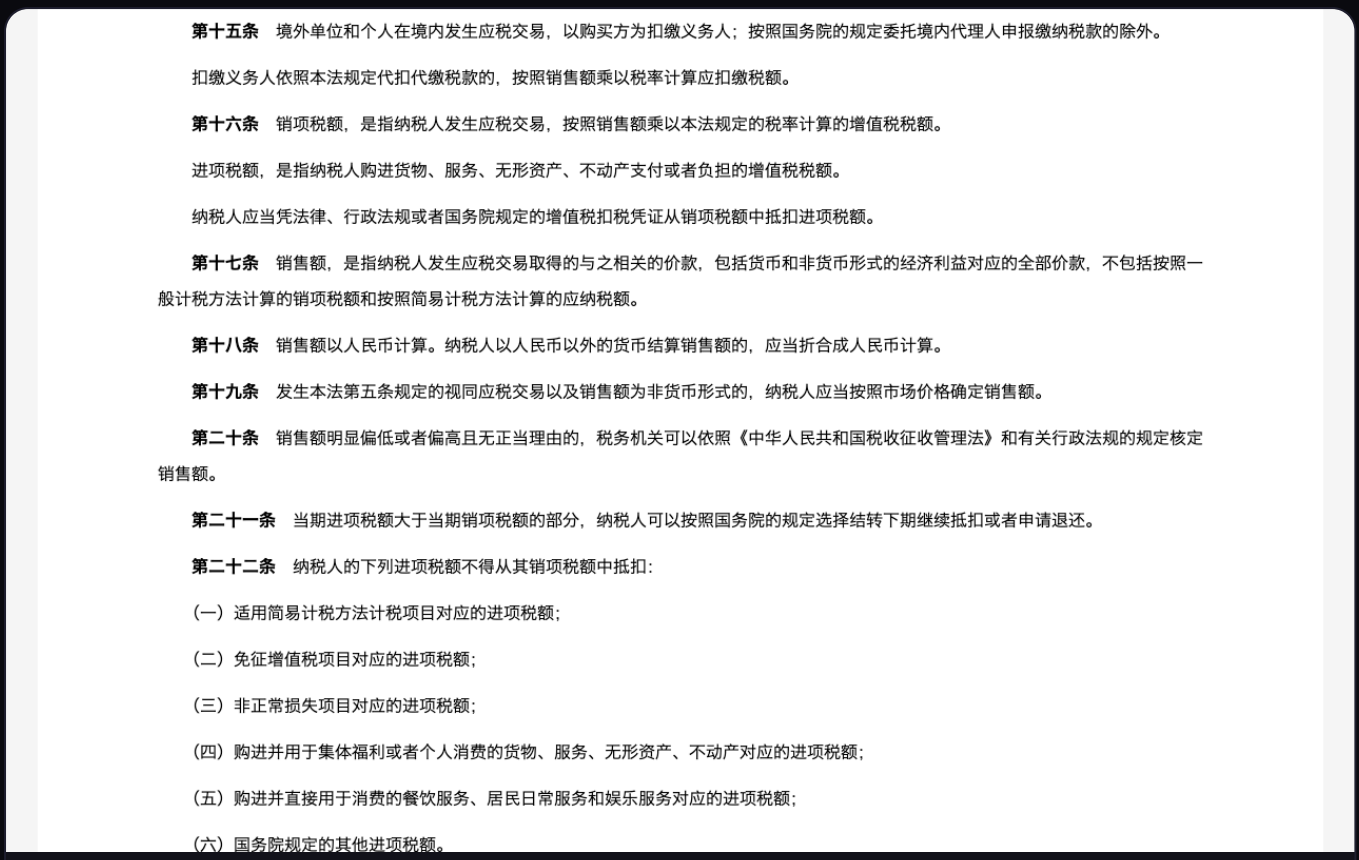
本体建得对不对，**靠什么判定?**

学术论文的做法是拿 F1 比对一个"标准答案本体"。可**企业没有标准答案**。 你得跟客户事先约定口径：比人工基线、看专家采纳率、还是算单据处理周期。 而且

说"餐饮服务都不能抵扣"，而这会让客户当场质疑你的专业性。

一个，业务方就不敢让它跑在真实流程上。

必须事先 — 事后再谈标准，一定谈不成。



第十五条 境外单位和个人在境内发生应税交易，以购买方为扣缴义务人；按照国务院的规定委托境内代理人申报缴纳税款的除外。

扣缴义务人依照本法规定代扣代缴税款的，按照销售额乘以税率计算应扣缴税额。

第十六条 销项税额，是指纳税人发生应税交易，按照销售额乘以本法规定的税率计算的增值税税额。

进项税额，是指纳税人购进货物、服务、无形资产、不动产支付或者负担的增值税税额。

纳税人应当凭法律、行政法规或者国务院规定的增值税扣税凭证从销项税额中抵扣进项税额。

第十七条 销售额，是指纳税人发生应税交易取得的与之相关的价款，包括货币和非货币形式的经济利益对应的全部价款，不包括按照一般计税方法计算的销项税额和按照简易计税方法计算的应纳税额。

第十八条 销售额以人民币计算。纳税人以人民币以外的货币结算销售额的，应当折合成人民币计算。

第十九条 发生本法第五条规定的视同应税交易以及销售额为非货币形式的，纳税人应当按照市场价格确定销售额。

第二十条 销售额明显偏低或者偏高且无正当理由的，税务机关可以依照《中华人民共和国税收征收管理法》和有关行政法规的规定核定销售额。

第二十一条 当期进项税额大于当期销项税额的部分，纳税人可以按照国务院的规定选择结转下期继续抵扣或者申请退还。

第二十二条 纳税人的下列进项税额不得从其销项税额中抵扣：

（一）适用简易计税方法计税项目对应的进项税额；

（二）免征增值税项目对应的进项税额；

（三）非正常损失项目对应的进项税额；

（四）购进并用于集体福利或者个人消费的货物、服务、无形资产、不动产对应的进项税额；

（五）购进并直接用于消费的餐饮服务、居民日常服务和娱乐服务对应的进项税额；

（六）国务院规定的其他进项税额。

直接用于消费的餐饮服务"。这五个字就是例外，转售时可以抵扣。来源：上海市税务局政策法规库" onclick="openLightbox(this)">

增值税法第二十二条六项不得抵扣情形 —— 第（五）项写的是"购进并**直接用于消费**的餐饮服务"。这五个字就是例外，转售时可以抵扣。来源：上海市税务局政策法规库

> 这三样填不了，不是因为算法不够好。是因为它们不在算法里，在人的脑子里。

TAKEAWAY

收尾

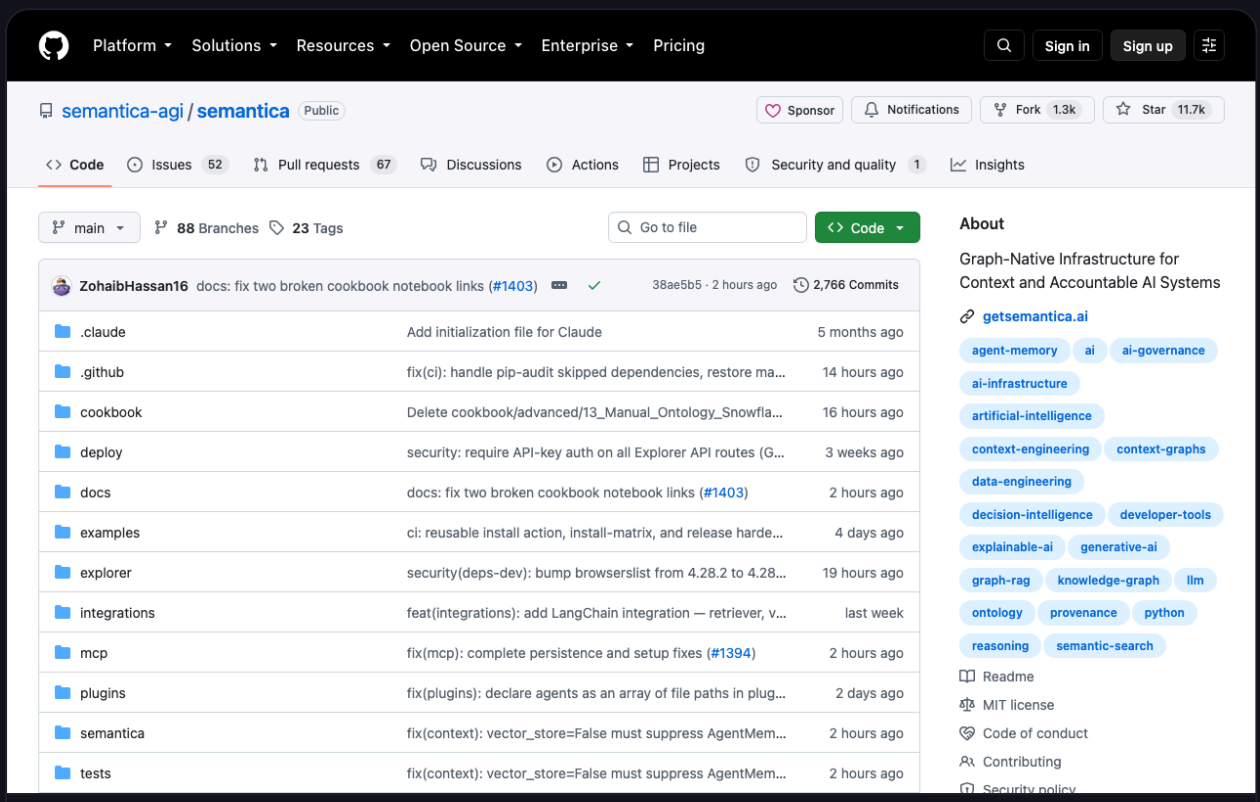
前五步是把知识捞出来，
第六步是确认它没错。

市面上绝大多数方案在第五步就交货了。
而企业真正付钱的，是第六步之后的东西。

下期：我把四条技术路线用同一份语料真跑了一遍

语料选的是**增值税进项抵扣**，全部来自国家税务总局和上海市税务局的公开网站。选它是因为《增值税法》今年 1 月 1 日才施行、《实施条例》去年 12 月 19 日才通过 — **模型训练数据里几乎没有，不能靠记忆蒙。**

跑出来的结果里：有一个**一万一千多星的项目在第一步就断了**，还有一个**流传两年的结论被一句 prompt 推翻了**。



就是这个 —— 11.7k star、MIT、两小时前还在提交，自称"Graph-Native Infrastructure for Context and Accountable AI Systems"

这六步我做成了一张自检清单 — 每一步该问什么、卡点在哪、什么信号说明这一步没过，最后还附了四条立项判断标准。需要的伙伴在评论区说一声。

论文依据：OntoEKG (arXiv:2602.01276) · Salovskii et al. (arXiv:2604.20795) · NeOn-GPT (ESWC 2024) · LLMs4OL Challenge @ ISWC 2025