从“念咒语”到“造世界”：提示工程退位，上下文工程登场

2025年8月10日　1 分钟阅读

#Context Engineering

> *从“念咒语”到“造世界”：提示工程退位，上下文工程登场*

一、热闹之后，真正的变革在哪里？

这几年，提示工程成了AI圈的“流量密码”。你几乎每天都能看到某个“万能提示词”，声称能把LLM变成律师、医生、程序员。表面上看，这像极了互联网早期的“SEO秘籍”：改几个词，性能翻一倍。

但行业里真正的变化，其实发生在另一个维度——上下文工程。简单说，提示工程是把一句话打磨到极致；上下文工程是把一个世界搭到位。前者像劝说，后者是供给。你不再押宝“神奇咒语”，而是系统性地构造AI思考所需的环境：事实、记忆、工具、流程、约束。这是从“舞台上的魔术表演”走向“工厂里的质量工程”的转型。

二、为什么“系统”才是主角？

我们先把视角拉长。任何一个可规模化的智能系统，都绕不开三件事：

> 输入要对：给的不是一堆原材料，而是加工好的半成品，乃至可直接装配的部件。
>
> 过程要稳：能抗扰动，换个说法不至于崩；能自检自修，出现偏差能往回拉。
>
> 输出要可控：有可预测性，有一致性，能复用到下一个场景。

提示工程的问题在于，它把复杂性推到一句话上，天然脆弱；而上下文工程用系统思维把复杂性拆解、摊平，然后用流程与工具稳住。Tobi Lütke、Karpathy、Harrison Chase的共识都指向这一点：

目录

文章信息

字数

阅读时间

发布时间

更新时间

标签

#Context Engineering

真正的核心是“上下文窗口里填的是什么”，以及“这些东西是如何被动态组织起来”的。

三、RAG只是“拼图的一块”

很多团队把RAG当成银弹：接了向量库、上了检索，感觉就有了“长记忆”和“最新知识”。其实这只是上下文工程的一个子模块。完整的拼图至少包括：

- 检索：找得到、找得准（RAG）
- 记忆：用户与任务的跨会话状态（短时+长时）
- 工具：API、数据库、代码执行的行动能力
- 编排：多步骤、多角色、多轮迭代的指令与控制流

把RAG升级为“动态上下文组装”才是正途：不是把文档扔进去，而是有目的地抽取、重写、排序、对齐，让模型“在正确的时间看到正确的事实”。

四、真正的痛点不是“AI忘性大”，而是“画像碎”

为什么很多AI看起来记不住你？不是因为模型“健忘”，而是你给它的“你”是碎的：电商有购物史，客服有工单，产品有行为埋点，CRM有画像，邮件里有上下文。每次对话都像在盲盒里抓一把。解决之道不是“把聊天记录加长”，而是构建随时间演化的统一用户画像——最好是时序知识图谱：谁、在何时、因为什么、做了什么决定、状态如何变化。只有当“你”被拼成一个连贯叙事，个性化才是真的。

五、没有共享上下文，多智能体只会各说各话

“AI团队”的迷思很常见：一个经理Agent拆分任务，几个工人Agent并行干活，听上去美。现实却常常翻车：背景是马里奥风，小鸟是另一套画风，最后拼不起来。这个教训很直接：协同任务的前提是共享上下文，一致的世界状态、一致的规范与约束、一致的验收标准。否则就是分布式瞎忙。

六、长上下文不是把料越多越

好，“中间失忆”很要命

上下文窗口做到了百万token，看起来“全喂给它”就万事大吉。但研究发现“lost-in-the-middle”现象很顽固：开头和结尾权重高，中间的信息易被忽视。因此，真正有效的做法是结构化上料：分块、提纲、摘要、锚点、引用、交叉验证；重要事实放前后，关键路径多次强化；对话中做“滑动窗口+关键帧”式的状态保持。信息不是越多越好，关键是密度、位置与重复策略。

七、从“技巧包”到“优化问题”

上下文工程的目标，是把信息“喂给模型”的效率最大化。这就像“摄影”：你拍的照片，是你“技巧包”里的“好照片”。但“好照片”的定义，是“能在有限的时间里，把所有重要信息都展示出来”。这就变成了一个“优化问题”：在给定时间窗口内，选择哪些信息最有价值，以最大化任务质量。

上下文工程正在被形式化：给定窗口约束、成本约束，求上下文组合函数的最优解，目标是最大化任务质量。信息论告诉我们要最大化互信息；贝叶斯给出了不确定性下的更新策略；工程上则落成一套“ROI驱动的上下文选择”：每个token都要算成本/收益，能不进窗口的就别挤。这一步很关键，它把“经验主义的摄影”变成了“参数化的光学”。

八、一个可落地的工程蓝图

把抽象落到工程抓手，我建议按“三支柱+两条线”的方式搭：

支柱A：信息获取

RAG：分层索引（段落/表格/代码）、领域适配的chunking、检索重排序（cross-encoder/ColBERT）

内部思维：Few-shot+思维链，必要时让模型先生成“任务草图/纲要”，再据此补料

动态组装：按任务schema拼装上下文：任务定义→证据包→工具说明→历史要点→验收标准

支柱B：信息加工

反“中间失忆”结构：摘要树（map-reduce）、提纲-证据-结论三段式、关键事实首尾锚定

自我精炼回路：草稿→评审→修订，多回合但设上限；给出评分rubric，显式对齐目标

支柱C：信息管理

记忆分层：短期对话态在会话内滚动；长期偏好与事实进向量库/知识图；建立"关键帧记忆"

上下文ROI：为每类信息设影响因子与老化函数；定期剪枝与归档

贯穿线1：工具与工作流

函数调用规范化（JSON schema），工具可观测（日志/回放）

工作流>单次推断：把复杂任务拆成确定性步骤+LLM步骤的流水线，失败可回滚

贯穿线2：共享上下文与治理

多Agent共享"黑板"：共享世界状态、接口契约、设计规范、验收清单

治理：提示版本化、数据血缘、评测集与回归测试，线上漂移监控

九、现实世界里的价值与边界

金融、医疗、制造、客服的案例已经证明：当你把"认知环境"搭好，AI的价值曲线不是线性的，而是折点上扬的。

边界也很清楚：没有统一画像的个性化是伪命题；没有共享上下文的多Agent是噪声放大器；不做ROI管理的长上下文是烧钱机。

十、我们每个人的新角色：认知环境的架构师

这场变革的本质，是人机分工的重画。过去你问得巧，模型答得妙；未来你搭得对，模型做得稳。你的工作不再是"写一句好提示"，而是"定义优化问题、设计变量与约束、构建数据与工具的可用边界"。说白了，就是给AI"装操作系统+上应用商店+配权限治理"。

一个形象的比喻：上下文工程就是给"尼奥"上传功夫。你不再给它一招一式的口令，而是把需要的场景、规则、素材、工具，一次性装到它的思维环境里，让它在这个世界中自洽地行动。

十一、展望：从会话到世界模型的跃迁

接下来三条值得押注的演化方向：

> 上下文的"世界模型化"：从静态片段到可查询、可推理的时序知识图与因果图；从"信息集合"升级为"可计算的世界状态"。
>
> 模型-工具-记忆的联合优化：不再把RAG、提示、调用当成后处理，而是训练时联合对齐（检索器、选择器、规划器端到端优化）。
>
> 评测范式升级：从静态问答分数到"任务成功率+成本+延迟+稳健性"的多目标指标；A/B与回归测试像CI/CD一样常态化。

结语

提示工程是把语言的边角打磨得更光；上下文工程是把智能的地基浇筑得更牢。前者带你进门，后者决定你能不能把厂子开起来、把质量做稳定。问题已不再是"AI能不能答对"，而是"我们能不能把一个足够完整、足够可控、足够经济的认知环境搭起来"。当你开始用ROI去度量每个token，用黑板共享去规范每个Agent，用世界状态去驱动每次推断，你就从使用者进化成了架构师。

这一步跨过去，你做的不仅是一套系统，而是一种未来的生产关系。