AI 开年炸文：你以为懂 AI，其实只是在背名词

原创 Shawn Shawn的想法 2026年2月7日 01:53 广东

好久没写长文了，到处出差。想写一篇大白话的，正常人类能看懂的“AI的前世今生”。

最近出差的路上，总是接收到很多的“噪音”，有来自投资人的，也有来自创业者的。噪音就是那些你也看不清楚、想不明白，看上去很有道理，但是仔细推敲下去有很多前置条件的观点。

而往往我们都忽略了常识的力量，忽略了事物本质的力量。用一句我常常挂在嘴边的话：“这玩意翻出朵花来，其本质是xxx。”，我善于去构建“不断更新的”结构化归纳。

但其实一开始我是反对归纳总结的，因为这会带来很多信息失真，你要归纳的前提是压缩，压缩到一个足够精简的总结（你看LLM中的Memory其实很多技术方式也是一个总结压缩，但这就会失真，损失信息）

但是如果这个归纳的只是骨架呢，即事物本质的属性，比如归纳的是：事物不变的维度、属性，这些都可以呀。在这之上去搭建骨肉，那是不是随着外部信息的迭代，自我认知的更新。这些都可以插上去当血肉。这样的失真是很小的，因为其骨架是正确的，无非肉少点，但其总体是不会出错的，这很有利于我们探索一些新事物。于是，我们就拥有了解析新学科的能力。

接下来是正文，那么AI前世今生也来自这样的“不断更新的”结构化归纳。

----------这是分割线----------

相信所有AI从业者、创业者、投资人都早已将“All your need is attention”这篇论文背得滚瓜烂熟了，我们也从这篇论文出发。

我们现在讲AI、讲AI应用、讲AI硬件、讲具身智能，其大前提是这里讨论的AI几乎就是生成式大模型。

那生成式大模型怎么搭建骨架，那就得回到大模型是啥。（很多人想，这不是废话吗，2026年了你还来科普啥是大模型？）非也非也，世事万物，万变不离其宗。大模型本质是一套基于海量数据预训练的算法建模，其工作的方式是，基于输入的数据，概率拟合生成输出的数据。这是其最本质、绝对不错的定义，我们的推理与归纳总结，应该由此展开骨架。

那么我们得出大模型的四大维度：数据、算法、输入、输出。接下来我们将已经发生的技术路演、产品思路往里面装。同时，让我们思考未来又会走向何处。

其一，数据

1，那么第一思考的角度应该是数据的类型。

从最开始的直接用文本训练，我们自然思考发现，如果后续可以用不同类型的数据训练呢，于是出现了图片训练、视频训练，但到了这里我们分成了2种技术路径。

要么先通过大模型外部工程能力去解析图片、视频，再转成文本，让大模型输出输入，这就是最常见的多模态。

要么，如果训练的数据本身就是多模态，是否可以直接去理解多模态的内容，而不需要工程化转录造成信息失真（例如录音转文本，其实是极难获取声音中情绪这个数据维度的），这种直接由多模态数据输入，输出也是多模态数据的范式，最早由马斯克提出，叫作“last to last”，也就是端到端。

那么我们自然思考起来，这个显而易见不失真、无延迟、更完美的数据是否有什么弊端呢，

当然有！

首先，数据的获取难度是极大的，端到端意味着同样的一个解析你需要有对应的端到端数据。例如，为什么之前王兴兴提到，谁第一个研发出机器人大模型，谁就可以得诺贝尔奖？因为比如“机器人抓握苹果”这么一个动作，你需要上亿次数据，那你又要做端到端机器人大模型，那就需要这个动作对应的机电参数，你去哪里弄？

这不比图片训练啊，“苹果”这种图片能找出无数张，你要“机器人抓握苹果”这么一个动作的机电参数是无解的，是无法想象的数据维度。

我能想到的解决方案有2种，其一找个机器人大数据工厂，每天无数机器人进行动作训练和数据提取。不然就只能用人造数据、模拟数据来做RL强化训练（也就是我编个数据给你用），但这带来就是这无法像LLM一样涌现出智能，机器人会很蠢。

当然，特斯拉做到了，他只用一个维度，就是摄像头，去端到端解析世界万物，理论上数据够多，是可以的，这需要海量汽车每天跑，每天收集数据，但这需要时间、需要车数量多。

所以大部分车厂变成了一个尴尬的地步，做VLM的端到端吧，没那么多数据。做非端到端的自动驾驶吧，corner case又太多，尤其对于很多汽车数量不够的厂商，使用人造数据、模拟数据就不可避免了。所以你看，context is everything（手动狗头），哈哈

那我们开始瞎想哈，端到端的优势是如此明显，训练的数据又是那么匮乏，而且只要是端到端就避不开物理世界的数据采集。所以会不会有AI硬件创业者这么思考，如果我做的device先不完全解决端到端的问题，我先用多模态顶着先用用，端到端的数据我先收集着，量变带来质变，最终成为端到端AI硬件？

这是啥，这是物理世界参数的数据化，是个好故事，应该挺多VC buy in的，但实现起来太难了，不好切，需要一定势能的人去做。

但聚焦到一个单一数据维度做端到端，是有可能的。别搞什么机器人大模型啊（机电数据作为训练类型那种），就用大家最常用的数据类型：声音、视频。已经够了哎。哦，不对，这就是特斯拉（二次手动狗头）。

但为啥特斯拉这么做还成了，第一性原理和极限思维很重要，就这事从根本逻辑上成立，然后拉到一个夸张的数量级是否能实现。如果可以，马斯克会去尝试的，这是很本质的思考方式不同。我也这么学着创业一路走来的，但还在努力中。

那未来的数据类型这条线将会走向何方呢，我还是认为在短期1-3年内，应该在相对垂类场景，去做可控的多模态转化，完全够用了，比如plaud、比如looki。

当然很重要的点是，这些垂类场景你不可能收集到，足够端到端预训练的数据规模。但垂类场景，只要熟悉业务know how，用一些多模态

转化技术足够了。你看plaud，谁会关心plaud录音的情绪？

以此类推，一秒视频分24帧解析的looki、桌面游戏陪伴、商务录音，都是不错的想法。还有一些比如可以记录睡眠的数据（呼吸声？血氧？）转成json文本，不用端到端，一样做够小场景用了。

再以此类推，做这些大垂类场景的数据采集器也是个不错的生意啊，比如大健康，采集了转成模型能理解的上下文，采集器卖给硬件厂商，也不错。太多啦，想不过来了，总之，专注到一些垂类做数据多模态挺好的，能真实解决一些问题。

但我觉得没那么有趣，就人总是想做点足够大的市场，影响足够多人的产品吧，趁年轻多试试吧。

不过安克和小米真应该系统化思考下这个方向，这可能是iot赛道的下一个重要故事没准。

2，第二应该思考的是数据的规模

过大的训练参数是会带来能力的加强，但是同样也会带来算力的要求过高、难以部署等问题。随着模型蒸馏的成熟、训练数据的干净度提升，以及算法本身能力的提升（按理这里应该和“算法篇”挂钩），但在这就一起写了吧。

出现了很多数十B甚至数B 参数的模型，比如千问开源的6b生图模型，就让我大为震撼，6b啊！6b它出个啥我都得夸他一句不错，结果实际体验下来真不错，赶上豆包生图早期版本了。

扯远了，那么我们为啥需要不同数据规模的模型呢？那就要回到小参数模型的特点。

其一，快，同样算力下，出token速度极快。由此展开，我们可以设想下，快有啥用？大用啊！如果生产速度大于消费速度呢？这不妥妥的楚门的世界吗，所以快就意味着有可能在某些场景写，用户消费的速度赶不上生成，用户产生一种我可以无限看专为我定制生成的内容了。

其二，垂直，首先训练数据小，或者蒸馏的弊端，小模型泛化能力是极其有限的，只能在某些特定场景去使用。

那什么场景是小算力，垂直的，且需要无限量内容供应的呢？坦白想好像没啥啊.....不过如果能将Gemini 3这种智能级别模型压缩到几十B，或许就真能出现全新的AI native device了，所以也许小参数模型的未来不在于垂类小算力场景，而在于能不能把目前最好的模型压缩到可以个人本地部署。

这是个软硬件一体的活，软硬件大厂应该投入精力去做的，华米OV，做超大参数的模型云端运行并不是个软硬件足够优雅的解决方案。做产品嘛，优雅很重要。但总体这应该是个研究方向，不适合做为创业方向。

其二，算法

我们讨论的一切算法的起始都是transformer架构，这是个相对单一的输入输出模型，人为可以控制的层级是相对有限的。这就导致后面出现了各种古早的提示词工程：

让AI先输出一段思考，在输出正文，来提示输出正确的概率，这叫啥，这叫思维链（Cot）；

先告诉AI一个示例，让AI仿照生成，这是啥，这叫小样本提示（Few Shot）；

告诉AI他需要扮演一个xxx的role,这是啥，这叫 cosplay（手动狗头三次）

扯远了，所以你们看，由于transformer架构的单线程、可控层级有限等问题，我们试图在算法层面直接解决这些看上去挺蠢的提示词，立马可以想到的，就有2个思路了：

1，我是否可以通过把Cot、Few Shot等提示词对应的完整按RL的方式训练进去，是不是我们以后在提示词层面就可以弱化很多了？好思路，但数据量太小了，直接训练效果不好，只能尝试微调了。所以你看，后面模型都出了很多小版本更新，这就是在对应小领域进行了一定优质数据集的微调。

2，我们提到之前的transformer架构是个相对单线程的模型，那如果在模型内部提前实现并行输出呢。我们提到工程化一般有两种，一种是模型内工程，一种是模型外工程。这属于模型内工程。这种在内部提前分化为不同节点分析也就是后面deepseek的moe（多专家）架构。

坦诚讲，其实算法层我并不算很专精，所以我更多会从一些产业化落地的方向去思考。

上述提到了一些算法的调优，但我们发现无外乎是模型内部的工程能力提升，比如增加思维链、增加多专家分析，其实这些事Agent也能做，但做在模型内部，可以做到上下文注意力的高度统一，不需要解决多智能体之间的记忆失真问题也挺好。

加上前文提到的小参数模型，我大致认为未来模型算法层面的方向在于2点：

其一，大量的模型内工程化能力提升，模型内嵌入记忆、思维链、角色等等。

其二，顶级模型的参数缩小，可以实现端侧的部署。（这个目前看好像还不明显，大部分模型厂还在大力出奇迹，认为大参数会出大东西。但起码国内的硬件厂商真的应该去重视才对）

这2点会直接导致创业生态产生巨大变化，我们应当做模型迭代我们更强的事，才有机会。

在这个技术路径迭代的范式下，我们创业能掌控的在大模型之外的，是2个东西：

Context：你模型再牛逼，也需要合适的上下文去驱动，要知道用户的上下文获取是困难的，我们无论用硬件、软件，都应该通过产品本身去驱动用户的上下文摄入，比如一个大健康穿戴硬件，就应该舒服的获取用户身体数据作为上下文给到大模型使用。用户在陪伴场景下，与自己OC角色的记忆。这些都算啊，大模型是无法舒舒服服获取这些上下文的，我们可以在产品侧构建用户心智，完成PMF。这就是机会。

在垂类场景下，完成业务闭环：比如法律咨询，大模型外挂了skill是能做，但他不能帮你联络某个律所给你打印合同吧？比如大健康陪伴，大模型再牛逼他不能根据你身体情况直接卖你蛋白粉配送到家吧？所以你看，在场景之下，利用大模型不能或者不会做的事完成业务最后一米的闭环，就是PMF。

其三，输入

看到这，很多朋友无语了，输入也能拎出来单独讲？当然可以，太有价值了。

首先回到大模型定义，输入甚至是最重要的，没输入，光一个模型是没有任何价值的。这就自然延伸出几个问题，输入什么？谁来输入？什么时候输入？在哪里输入？我们一个一个思考。

输入什么：这需要收敛到一个核心命题：我们需要输出什么样的内容。所以抛开输出谈输入是耍流氓的。那对于我们希望得到的输出来说，我们需要与之匹配的输入（也就是上下文）。要明白对于输入来说，是一次性灌注的，然后再一起输出，那在很多场景下，我们首先需要通过外部的能力去筛选所需的上下文加入到输入中。

如果是客服场景，对于很多用户常问的问题，我们可以外挂一个知识库，可以通过各种方法去搜索，进而找到对应的答案，然后把答案塞到上下文中，这就是RAG（搜索增强生成）。

古早的时候还叫做embbeding（向量化嵌入），这是一种向量化检索的搜索方法，例如用户问了“草莓冰淇淋好吃吗？”，通过将这段内容向量化以后，与知识库中的内容做向量化匹配，寻求cos夹角最小的就是对应的内容。

所以embbeding只是数据存储和收集的一种方式，因为天生很契合大模型，被open AI首次提出并使用。但后面大家发现，其实数据的存储、排序、查询是有很多方法的，这就引申出了很多技术路线，然后统称为RAG。

那如果是对话场景呢？陪伴场景呢？AI需要记住你是谁啊，要知道AI就只有单侧输入这一个层面，哪有真正的记忆。那就要在模型之外做记忆体系，问题怎么去定义记忆是什么？对于用户那么多的内容，应该选哪些存到记忆中。总体而言是复杂的，目前市面上有2种主流的记忆技术路径，要么对历史上下文做总结，然后不断更新这个总结。

要么都存在下来，基于用户提问再去查找对应的记忆。也有两种融合的。

总之记忆工程是个复杂的、多种技术路径的事儿，但坦率讲这不是一个很好的创业方向，因为这玩意儿太底层了，几乎所有模型都会自己做，对于创业者来说实在难以在大后期生存下来。

那有了记忆体系，我们就需要在需要调用记忆的时候，往上下文里面去塞无论是总结的还是查找出来的记忆内容。

同理，如果是AIGC领域呢？无论是漫剧生成、AI设计、Vibe coding，都需要在各自对应的场景下，去在上下文中插入适合的内容。可能是首尾帧图片、可能是参考示意图、可能是代码。

那如果是物理世界呢？同样的，录音、帧截图、录像都可以塞到上下文中。

但是大家会为这些形式包装很多酷炫的名词，但聪明如你，一定发现了。到头来最终和大模型产生交互的那个点，就是一段精选过的内容，塞到了输入中作为上下文。

是的，所以这一切，现在有一个统一的名字“上下文工程”。我常常挂在嘴边一句话，“context is everything”,我们创业者的机会也在这个context（上下文）中。因为只有我们自己清楚具体的用户场景，我们是否能够舒服的拿到context、是否可以选择这个场景下合适的context、以及怎么用这个context将成为我们极强的壁垒。

例如Plaud，至今还有很多投资人和创业者认为这玩意为啥不直接用手机打开大模型APP来替代。想不明白是因为他们不理解context的重要

性。Plaud可以让商务用户一键录音获取context，并将这部分context进行符合用户场景的利用（转录、总结、提炼）。

同样，还有许多领域当大家具备对行业的深度理解的时候，我们可以比通用大模型让用户更舒服的给我们context，并且我们对于这个场景下的context利用到了极致。这种体验是通用大模型无法比拟的。

谁来输入：一开始大家下意识认为，输入应该来自于人类打字。但其实从原理来说，它可以来自任何东西，可以来自其他AI、来自API调用结果、来自触发器等等。到这里，我们终于触及到了大模型的核心，即可以通过**首尾相连+外部调用**，实现把智能装进工作流中。这就表明，AI不是泡沫，它一定是生产力级别的变革。而任何的生产力级别变革都会成为全社会的变革。

从这个原理出发，我们第一次将“智能”这件事变成一个单元组件，在脱离开人类输入后，AI可以由其他AI输入控制、由很多不知道是什么的玩意控制。在具体场景下，这种智能化的改变会渗透到方方面面，理论上所有需要低端智能的场景都可以被替代，比如文稿审理、轻度咨询行业等等。

什么时候输入：大部分人都会认为我们需要一些触发输入的时机，但这其实都是我们人类自己的判断标准。理论上AI可以无限连轴转，不断进行输入-输出的循环，直到完成某些任务。这种自循环规划能力是区别于RL的，或者说这是一个强化级别的RL。

但不难发现，这表明任何一个复杂的任务需要海量的token和巨长的时间。那么这样的时间成本和算力成本，究竟什么的场景才有可能使用？简单来说只有那些本身任务目标非常贵重的情况，且可以接受异步计算的场景。比如我的本科论文！可恨当年没有AI，不然我高低充

钱让AI帮我完成。所以AI for science是个挺好的赛道，真的有研究背景的同学，完全可以尝试下。

在哪里输入：这是个很有意思的话题，很多时候收集输入信息对于大部分用户来说是挺困难的，在具体业务场景下，比如一个大健康场景，你让用户给你他的心跳、血压、血氧，还是挺有挑战的，所以我们发现如果在某些垂类场景下，可以通过硬件、APP、传感器等等来获取上下文。这也是通过大模型做不到的，他只能做基座，最终从哪里找垂类用户场景的上下文还需要我们自己去解决，是否解决的优雅舒服，那就是极强的壁垒。再次夸下Plaud，做的真的不错。

写了这么多，其实也没写完，但是万变不离其宗，对于输入（上下文）的说到这里就够了，大家最终只要明白，不管三七二十一，和模型交互的最终窗口不过就是一段经过我们选择的内容就行了。选择什么内容、怎么获取这些内容、按照什么方式塞到输入中，这就是功力了了。

其四，输出

我们前文说到，这是第一次将“智能”这件事变成一个单元组件，那对于一个“智能”的输出我们可以很自然的从一些不同维度来思考。

其一，输出类型

谁说的输出的必须是直接交付给用户的内容呢？例如文字、图片、视频。不不不，我们完全可以放开想象力，输出的可以是json的数据格式，这样就可以实现对于外部软件功能的调用。也可以是对于不同AI的任务规划，这不就实现了任务规划的能力吗？

所以输出的类型应该随着具体的使用场景做变化，做漫剧的，可以是分镜稿、运镜描述，对于大健康赛道，可以是推荐的蛋白粉，最好来

个sku编码，直接发货送到家，这够智能了吧。还有很多场景，当我们放开思维，去解放AI本身“智能”的能力的时候。我们就可以定义很多输出的看类型。

其二，输出给谁？

这不是废话，除了直接输出交付结果外。AI的输出也可以是其他AI，这种模型可以实现自主的智能涌现与规划。再结合前文提到的各种用法，这就变成了现在的Agent 工程。到这里，AI已经作为一种原子级的组件可以去构建种种具体应用场景的智能涌现优化，只有想不到，没有做不到，当然现在的token 费用和时间问题依旧没有解决。

但从长远来看，这都会得到解决，别问我为啥能解决，一个行业涌进全世界最多的钱和人才的时候，它不行也会变得行，因为世间万物都是人类所创造的。

好啦～到这里，回顾全文。我们清楚知道了大模型的底层骨架，并在上面已经新增了许多骨架。现在我们来对现在新出现的概念与思路进行归纳。

Tools:基于外部软件能力的输入类型，当我们需要某类输入数据但是大模型本身不善于去生成，例如计算器，我们完全可以在识别到需要算数的时候，去调用一个计算器来获取结果。

MCP：当输入的数据类型如此丰富后，各大模型厂商是否希望存在一种范式，去规范与大模型的通讯，当然，这就是MCP，一种定义清晰的调用方式。

Skill：当单一大模型场景对于一些常用的场景，是否可以将该场景所需的所有工作打包起来，去让模型在输出的时候去调用？例如把查到

天气温度很高然后自动调低家里空调温度封装为一个Skill，类似这样去让开发者开发无数个封装组件，然后给个新名字Skill.

Agent: 当在输入侧、输出侧部分或者全部使用了我上文提到的一些能力，进而实现智能的自主化、外部工具的调用、直接的结果交付。这一整套基于具体场景的AI工程，就称之为Agent。

到这里大家终于可以不用再担心fomo或者被别人装x到自己看不懂了，其实AI的底层逻辑是非常简单的，它是一个可以单元化的“智能”，且只有输出和输出2个维度。希望这篇文章能为很多AI的创业者、投资人、从业者带来清晰的、高效的AI框架。

最后安利我们公司，Noonwake.AI，我们是一家以全球不同文化为切入点，探索AI陪伴与AIGC的AI应用公司。目前产品有：万象有灵app（国内）、Starot app（欧美）。后续还会做其他的文化地区。

欢迎交流，这是Shawn新年的第一篇内容。**全文手搓，无AI。**

修改于2026年2月10日