



观点与分析

让舞蹈停下来的蜜蜂

从蜂群智能的秘密到沙特 Kawn 实验室：为什么专业化的小头脑，或许能创造巨头无法独力完成的未来？

2012年，美国生物学家托马斯·D·西利（Thomas D. Seeley）及其同事在 *Science* 发表了一项引人入胜的发现，揭示蜂群如何作出最重要的决定之一：选择新家。

侦察蜂飞出去寻找适合筑巢的空腔。发现有希望的地点后，它会返回蜂群，以摇摆舞（Waggle Dance）传递位置信息。地点越理想，招募同伴的舞蹈就越积极，也会有更多蜜蜂亲自前往考察。

蜂巢里没有绘制蓝图的建筑师，没有比较方案的中央委员会，也没有下令迁徙的蜂王。

然而，成千上万只蜜蜂却能完成极其复杂的集体选择。

但西利的发现不只关乎这种舞蹈。

研究人员还观察到一个更令人惊讶的行为：一些蜜蜂不只是为自己的选择争取支持，还会向正在宣传竞争地点的蜜蜂发出停止信号（Stop Signals）。

神经科学把这种机制称为交叉抑制（Cross-Inhibition）。它能帮助群体打破相近选项之间的僵局，避免竞争无休止地延续。

仿佛自然早已明白：智能不仅需要有人说“我找到答案了”，也需要有人知道何时该说“别再沿着这条路走下去”。

研究人员在 *Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms* 一文中记录了这一现象。

这个细小的生物学观察，引出了一个远比蜜蜂生活更宏大的问题。

个体能力有限的生物，如何在没有任何一个成员掌握全局的情况下，产生高度复杂的集体行为？

这个问题，会不会正是通往人工智能未来的一把钥匙？

过去几年，我们习惯从规模来理解语言模型的发展：数十亿参数（Parameters）、更长的上下文窗口（Context Window）、推理能力（Reasoning）、通用人工智能（Artificial General Intelligence, AGI）的目标，以及随后出现的超级人工智能（Artificial Superintelligence, ASI）愿景。

这场竞赛有时像是在试图创造一种超级生物：读过所有书，精通所有语言，理解所有职业，回答所有问题。

但自然比我们的实验室早了数百万年试验智能系统，它提供了另一种哲学。

群体要变得更聪明，并不要求每一个个体都变得更聪明。

有时，真正的才智在于设计小头脑之间的关系，而不是把一个头脑无限放大。

不需要云端的机器人

2026年8月11日，Raspberry Pi 发布了一篇技术文章。乍看之下，它不如那些巨型模型的发布消息引人注目。

文章展示了如何在 Raspberry Pi 5 上通过 LiteRT 运行 Google Gemma 系列模型，其中还包括一个名为 Reachy Mini 的小型机器人实验。它能接收声音和视觉信号，并在本地作出响应，无须持续连接远端计算中心。

在公开的性能测试中，只有2.7亿参数的 Gemma 3 270M 通过 LiteRT-LM 实现了每秒约22.6个词元的解码速度；模型文件大小为278MB，测试中的峰值内存占用约为680MB。

多模态机器人实验本身使用的是该系列的另一款模型 Gemma 4 E2B；音频、图像和语言处理则由系统中的不同专业组件分工完成。

详细内容见[Raspberry Pi 官方技术文章](#)。

一款2.7亿参数的模型不会在科学难题上胜过最强大的通用模型，这个小机器人也无法在所有任务上同数据中心竞争。

但真正重要的消息在别处。

几年前还必须依赖远程服务器的某些能力，如今已经可以在一台掌心大小的计算机上运行。

更重要的是，这套系统并不要求一个模型完成所有工作。视觉由一个组件处理，语音识别交给另一个组件，语言模型负责决定如何回应，最后再由文本转语音系统输出声音。

四种专业能力汇合在一起，形成一种对用户而言自然连贯的行为。

2026年6月，Apple 已经发布第三代 Apple Foundation Models，其中包括设备端模型（On-Device Models），例如约30亿参数的 AFM 3 Core，也包括供其私有服务器使用的模型。这并不是抛弃云端，而是在设备与服务器之间重新审慎分配工作。[Apple Machine Learning Research](#)

把这两条消息放在一起，我看到的不只是模型缩小所取得的进展。

我看到的是工程问题本身正在转向。

我们不再只问如何把每项任务都交给可用的最大模型，而会越来越多地问：完成这项任务，最少需要多少智能？它又应该在哪里运行？

答案可能在手机里，在工业设备上，在机构自己的服务器中，也可能在确有必要时交给全球性的大模型。

目标不是让其中一个位置战胜其他位置，而是能够在它们之间作出明智选择。

为什么蚁群不需要首席执行官？

早在人工智能出现之前，动物世界就已经面对这样的问题：如何协调数量庞大、个体能力有限的生物。

蚁群怎样找到食物？怎样组织数千只工蚁的行动？当一条路径中断或新的食物来源出现时，它又怎样适应？

1959年，法国昆虫学家皮埃尔-保罗·格拉塞（Pierre-Paul Grassé）提出了一个后来在复杂系统研究中十分重要的概念：共迹效应（Stigmergy）。

它的核心是：个体无须直接向其他成员发出指令，也能影响它们的行为。

一只蚂蚁在路上留下化学痕迹，其他蚂蚁对此作出反应，而它们的行动又会改变各条路径上痕迹的强度。这些痕迹还会随时间和环境条件而变化。

大量局部互动由此形成有序模式。

单只蚂蚁并没有一张显示蚁群全貌的地图，但整个蚁群能够完成远超个体能力的工作。

研究人员把这种现象称为集体智能（Collective Intelligence）；如果某种性质只在群体层面出现，而不以同样形式存在于个体身上，则可称为涌现行为（Emergent Behavior）。

这些思想并非转瞬即逝的文学隐喻。它们进入了计算机科学和群体机器人（Swarm Robotics）领域，也促进了优化（Optimization）、路径搜索和任务分配方法的发展。

其中最知名的例子之一，是与研究人员马可·多里戈（Marco Dorigo）密切相关的蚁群优化算法（Ant Colony Optimization）。

我并不是说神经网络应该逐字照搬蚁群；生物学隐喻并不是一张可以直接施工的工程图纸。

但其中更深层的原则值得思考：

只要协作方式足够出色，强大的能力就可能从有限的单元中涌现。

也正因此，一项沙特实践格外吸引我。与其把它放进传统的模型规模竞赛中，我更愿意透过生命系统的视角来理解它。

它就是 Misraj AI 及其研究实验室 Kawn Lab 的实践。

Misraj：当目标不是一个无所不知的头脑

Misraj 的方向吸引我的，不只是它拥有一款小型阿拉伯语语言模型。

更关键的是，它已经开始构建一组专业化能力：每一项都有明确职能、知识领域，以及与任务性质相匹配的训练和评估方法。

这个家族的中心是 Kuwain 1.5B。它是一款拥有15亿参数的小型语言模型（Small Language Model, SLM），主要被用作进一步开发专业模型的基础。

但只有离开通用模型，看看它生出的各个分支，这个家族的故事才真正变得有趣。

其中有 **Mutarjim**：一款构建在 Kuwain 之上、专注阿拉伯语与英语互译的模型。

在研究人员发布的[2025年 arXiv 预印本](#)中，他们报告 Mutarjim 在特定测试里超过了规模大得多的模型，其中包括在 Tarjama-25 基准的英译阿任务上超过 GPT-4o mini。

这并不意味着它在所有语言和所有任务上都优于每一款 GPT 模型。

它展示的是一个重要的技术悖论：在明确限定的测试范围内，一款小型专业模型可以超过一款规模更大、在许多其他领域能力更强的通用商业模型。

接下来是 **Sadeed**，它专门处理阿拉伯语加元音符号（Arabic Diacritization）。

不了解阿拉伯语复杂性的人，可能会把加元音符号视为一项边缘任务。

但“علم”一词既可能表示“知识”，也可能表示“旗帜”，还可能是“知道”这个动词的一种形式。系统不能只凭字母外形选择正确读法；它必须理解句子结构、上下文以及词语之间的关系。

在古典和文学文本中，一个细小的元音符号就可能改变词语的语法功能，或改变整个短语的含义。

因此，加元音符号不是给文字添上装饰，而是一项同时涉及形态分析（Morphological Analysis）、句法分析（Syntactic Analysis）和语义消歧（Semantic Disambiguation）的任务。

在 Sadeed 预印本 中，团队报告使用了经过精心整理的阿拉伯语数据进行专业训练，并提出 SadeedDiac-25 评测基准，以比较不同类型文本上的加符表现。

随后是 Lahjawi 系列，用于阿拉伯语方言之间的翻译，以及把方言转换为现代标准阿拉伯语（Modern Standard Arabic, MSA）。

这与英阿翻译是不同的专业方向。各地方言不只词汇不同，句法结构、表达方式、社会含义也有差异；同一个词还可能因国家和语境不同而改变含义。

Nabr 则是一款面向《古兰经》诵读的专业语音模型，用于识别诵读内容，并检测部分发音和诵读规则错误（Tajweed-Aware Mispronunciation Detection）。

它不必充当一名通用助手，回答罗马帝国史，也不需要分析公司预算。

它面对的是一个高度专业化的声音领域，其中涉及延长音、鼻音、qalqalah（回弹音）、强调发音和诵读表现等特征。

在另一个方向上，Baseer 是一款拥有30亿参数的视觉语言模型（Vision-Language Model, VLM），专门理解阿拉伯语文档，在保留页面视觉结构的同时，将其中内容转换为结构化文本。

接下来这个家族，在我看来掌握着未来最重要的钥匙之一：嵌入模型（Embedding Models）。

其中最突出的包括拥有3亿参数的 Kawn Embed Light，以及各有1.39亿参数的 Kawn Embed Islamic 和 Kawn Embed Medical；它们分别面向不同领域的知识检索。

这些模型并不充当文章作者或聊天机器人。它们把文本转化为数学表示，即向量嵌入（Vector Embeddings），从而帮助计算机在措辞不同的情况下，仍能找到语义上最接近需求的段落。

据 Misraj 官网介绍，其研究人员在 LaTell 2026 展示的工作，提出通过多教师知识蒸馏（Multi-Teacher Knowledge Distillation）来实现阿拉伯语嵌入模型的专业化。

至此，一幅几乎与本文开头相呼应的图景出现了。

这个家族不要求每个成员知道一切。

一个模型负责聆听，一个负责翻译，第三个添加元音符号，第四个理解扫描页面，第五个则从数百万份文档中寻找语义。

知识在这里被分配，并不是因为团队无法造出一个包办一切的模型，而是因为在服务明确任务的系统中，专业化本身可能就是更好的工程决策。

在我看来，这是 Misraj 工作中最值得关注的思想之一。

价值不在于每个小模型都能击败所有巨头，而在于一套整合系统能够判断该由谁接手任务，以及何时把工作转交给另一个模型。

当模型组成一支工作团队

不妨设想一个贴近日常生活的场景。

一位母亲坐在家中，面前有学校文件和一份含有英文术语的报告；孩子想听一段带有阿拉伯语元音符号的文本；学校还发来一封语言正式、需要解释的通知。

按传统使用方式，她可以把所有材料都交给一个大型语言助手，请它翻译报告、提取信息、添加元音符号并解释指令。

这个助手也许能很好地完成任务。

但我们也可以设想另一种设计：它不假定所有问题都必须调用同一个模型。

如果纸张是扫描图像，就由 Baseer 这样的文档理解模型提取文字和结构。

如果文本需要翻译，任务就转给 Mutarjim。

如果需要为阿拉伯语文本添加元音符号以便阅读，就调用 Sadeed。

如果问题涉及特定文档中某句话的含义，则先由 Kawn Embed Light 这样的嵌入模型检索相关段落，再由合适的语言组件依据这些材料组织答案。

不必每次都运行全部模型。有些请求只需要一个模型，有些需要一条短链，还有些适合并行处理。

根据具体设计，这种结构可以涉及模型路由（Model Routing）或模型级联（Model Cascading）；如果再加入决策、工具使用和工作管理能力，它也可能进入多智能体系统（Multi-Agent Systems）的范畴。

它并不等同于单个神经网络内部的专家混合架构（Mixture of Experts, MoE）。后者激活的是同一个模型中选定的部分，而这里讨论的是功能不同、彼此独立的模型之间如何协调。

这种区别对工程师很重要，但道理并不难理解。

你不会请眼科医生修空调，也不会因为专业译者和注册会计师都受过教育，就要求译者代替会计师工作。

那么，我们为什么有时仍坚持让一个语言模型承担所有事情？

2023年，斯坦福研究人员 Lingjiao Chen、Matei Zaharia 和 James Zou 等发布了 [*FrugalGPT* 预印本](#)，研究如何在不同模型之间路由请求，以降低推理成本，同时在特定测试中保持性能水平。

随后，发表于 ICLR 2025 的 *RouteLLM* 研究了如何利用偏好数据学习为请求选择合适的模型（Learned Model Routing）。

这些研究让我欣赏的一点，是它们把一部分智能移到了模型之上的层次。

与其给每个模型都增添新知识，我们可以通过学习究竟应该使用哪一个模型来改进整个系统。

在这种转变中，善于在模型之间分配任务，可能会在工程价值上与善于开发某一个模型同等重要。

甚至有些我们以为需要更强大头脑的问题，真正需要的只是更好的工作分配。

设计方式	在哪里作出选择?	哪些部分协作?
模型路由 (Model Routing)	根据任务，在独立模型之间选择	不同专业能力
模型级联 (Model Cascading)	按需经过连续阶段	依次接手任务的模型
混合专家 (Mixture of Experts)	在单个模型架构内部	网络中被选中的部分

依据本节内容整理的概念比较，并非性能测试。

智能体或许能拥有的最大本领：知道何时停止

这里，我又回到托马斯·西利笔下的蜜蜂。

蜂群能够形成一致决定，不只是因为蜜蜂会提出新地点，也因为系统中存在一些机制，阻止较弱的选项持续下去，并帮助群体作出选择。

这让我想到一个许多智能体（AI Agents）拥护者容易忽视的问题。

把一项任务分给十个能力较弱的模型，并不会仅仅因为参与者变多，就让整个系统自然变强。

错误可能在模型之间传递：第一个模型从错误解释开始，第二个在它之上继续推演，第三个又自信地重新表述，最终得到一个极具说服力、却从第一步起就错了的答案。

因此，协作工程并不止于挑选专业模型。

它还需要不确定性估计（Uncertainty Estimation）、置信度校准（Confidence Calibration）、分布外检测（Out-of-Distribution Detection）和输出验证（Output Verification）。

当小模型无法达到所需的置信水平时，系统应当知道如何把任务升级给更大的模型（Model Escalation），或交由人工复核（Human-in-the-Loop）。

从工程角度说，系统必须具备拒答能力（Abstention），而不能把回答视为一项无论疑虑多大都必须履行的神圣义务。

在医疗、金融和法律机构中，这种能力可能比文采或文本生成速度更重要。

能够识别文档不完整或检索信息不足的模型，比一个凭空写出流畅答案的模型更有用。

我设想，下一代专业智能体在这种判断的质量上展开竞争，会像它们在答案质量上的竞争一样激烈。

问题不只是：智能体能不能完成任务？

还包括：它是否知道该调用哪个模型、该验证什么、何时求助，以及何时停止？

智能由此不再只是说话的能力，而成为管理知识及其边界的能力。

有时，团队中最聪明的成员，正是那个阻止团队犯下一个错误的人。

最快的头脑不一定是最大的

谈论小模型的效率时，我们很容易落入另一种简单化：以为参数更少，就必然更快、更节能。

但计算系统的工程远比这复杂。

性能取决于模型架构（Model Architecture）、权重的表示方式、推理运行时（Inference Runtime）的效率、输入与输出规模、内存速度、处理器类型，以及加载模型并让它常驻内存的成本。

通过量化（Quantization），可以减少表示权重所需的位数，例如从16位的 FP16 转为4位的 INT4，从而大幅降低权重所需的存储空间。

理论上，一款拥有1.39亿参数的模型，以4位存储权重约需70MB；这还没有计入量化元数据、临时内存和运行环境开销。

一款15亿参数的模型，在同样精度下，仅权重就可能需要约750MB，之后还要加上其他内存需求。

因此，认真考虑在手机和边缘设备（Edge Devices）上运行这类模型是合理的；但这并不意味着 Misraj 目录里的每一款模型，都已经在某一部手机或 Raspberry Pi 上接受测试并证明了性能。

能够加载模型，不等于能够以合适的速度运行它。

专业化还可能减少模型生成长篇回答或处理巨大上下文的需要，因此即使模型规模没有明显变化，也能改善整项任务的成本。

发表于 IJCAI 2026 的研究 *Mapping the Efficiency Landscape of Small Language Models* 考察了70多款小型模型。研究人员发现，更小的模型并不一定在每种情况下都更高效；能耗与性能之间的关系，也远比按参数量排序复杂。

这个结果很重要。

更聪明的衡量方式，不只是每秒生成多少词元（Tokens per Second），也不只是每个词元消耗多少焦耳（Joules per Token）。

更贴近个人和机构实际生活的尺度是：系统需要花多长时间、消耗多少资源，才能正确而完整地完成一项任务？

一款小模型在自由文本生成上可能很慢，却非常擅长高效地给短句分类。更大的模型也可能只需更少步骤就能解决复杂问题，因而反倒是更经济的选择。

工程智慧不在于永远使用最小的模型。

它在于：在准确性、时间和隐私的约束内，选择能以最低总成本实现所需结果的模型。

这仿佛是在重新发现自然界的一条古老原则：群落不会把所有成员都派到同一朵花上，也不会为每项任务分配同样的资源。

当手机成为智能发生之地，而不只是通往智能的窗口

我认为，未来几年一项重要的社会变化，不会发生在大型技术发布会的屏幕上，而会发生在我们早已习以为常的日常物件中。

手机、汽车、家用医疗设备、学校电脑和工厂摄像头，都可能获得专业化智能，并在靠近所需数据的地方运行。

设备并不需要掌握全世界的知识。

一位母亲的手机也许只需要一个在本地识别阿拉伯语语音的模型、一个帮助孩子阅读文本的模型，以及一个整理文档的模型，而无须把她生活中的每个细节都发送到远端服务器。

工厂也许需要一个从维护报告中提取信息的模型、一个给警报分类的模型，以及一个检索操作说明的模型，同时让设备数据始终留在工厂网络内部（On-Premises Deployment）。

在学校里，一个专业模型可以帮助纠正阅读，另一个可以简化文本，第三个则可以在明确的数据保护政策下跟踪学生的学习进展。

这并不是要取代教师、医生或工程师。

它是为了减轻那些占用他们时间的琐碎工作，并让某些能力即使在网络较弱或中断时，也能在本地继续可用。

一个重要的可能结果是：隐私会从附加服务变成设计本身的属性。

如果数据从一开始就不离开设备，那么与传输数据和在外部存储数据有关的相当一部分风险就会降低；当然，设备本身、本地模型及其使用日志仍然需要保护。

在机构内部，本地推理（Local Inference）和物理隔离网络部署（Air-Gapped Deployment），为那些难以通过公共云服务提供的应用打开了空间。

在这里，我看到了阿拉伯语专业模型家族的一项重要优势。

与其让一个巨型通用模型处理每项请求，不如让小模型在机构边界内运行，把复杂任务留给更大的内部服务器，并只在政策允许且确有必要时，把任务升级到外部服务。

Misraj 已经展示了用于构建其模型应用的产品，例如用来编排处理与路由流程的 Seamless API，以及面向运营型智能体的 Workforces，并提供本地和企业部署选项。

这些都是可以发展为多模型系统的架构积木。至于每一种组合相较替代方案究竟有多大优势，仍取决于具体任务、硬件和实际运行测试。

或许最美好的人本收益，是用户越来越不需要知道人工智能叫什么名字。

一些任务会隐入手机或应用之中：需要翻译的地方，文字自然被翻译；需要理解的地方，文档自然被理解；出现错误的地方，文本自然得到纠正。

人工智能成熟的一个标志，或许是我们每次从中受益时，都不再觉得自己正在“使用人工智能”。

就像今天每次开灯时，我们已经不会特意提起电。

当阿拉伯语成为一种工程优势

小型专业模型还有另一个超出内存和能耗的维度。

阿拉伯语在形态、方言和语义上的多样性，使其难点并不会均匀分布在所有任务中。

检索阿拉伯语医疗文档，与识别诵读错误不同；它又不同于海湾方言与马格里布方言之间的翻译，也不同于从扫描文档中提取表格。

通用模型必须在一套能力中覆盖这整片宽广的任务谱。

专业化则允许把数据、训练和评估直接对准明确的问题。

以 Kawn Embed Medical 为例。

在医疗文档中，患者可能用口语描述症状，而临床记录却使用专业术语表达同一概念。专业嵌入模型的作用，是拉近这些不同表达在语义空间中的距离，帮助系统找到相关信息。

它不会仅仅因为找到了合适的文本就诊断疾病；它承担的是医疗应用所需的一项精准检索功能。

Kawn Embed Islamic 面对的则是另一个问题：把查询与《古兰经》、圣训、经注和教法中的相关文本连接起来，同时保留来源归属的重要性。

两款模型的规模可能相近，但训练数据、语义空间和成功标准并不相同。

我认为，这种灵活性为一种可以称作超本地化专业人工智能（Hyperlocal Specialized AI）的方向打开了广阔空间。

这类模型理解特定社群的方言、专业表达和文档，却无须重新学习全世界的所有知识。

在各自专长中，它们可能比通用模型发展得更快，因为更新狭窄任务的数据并评估其影响，有时比改动一个全面的知识系统更容易管理。

这并不意味着巨型模型已经失去用途。

先进的通用模型仍会在复杂推理、陌生情境、跨知识领域连接，以及为困难案例担任监督者或后备处理器等方面发挥重要作用。

但在许多设计中，它们可能会变成只有在必要时才调用的能力，而不是面对最简单的事情也自动启动的能力。

这是设计哲学上的重大变化。

一种只在需要时才为天才付费的经济

国际能源署（International Energy Agency, IEA）在能源与人工智能报告中估计，全球数据中心2024年的用电量约为415太瓦时；它在后续更新中预计，到2030年这一数字将接近950太瓦时。

人工智能并不是数据中心耗电的唯一原因，但它已经成为用电需求增长最重要的驱动力之一。

随着模型应用扩展到各种活动，一个新问题随之出现。

如果数百万人每天都用人工智能处理小任务，那么，把每一项任务都交给能力最强、资源消耗也最大的模型，在计算上真的明智吗？

这就像用一架货运飞机运送一封纸质信件。

信件也许能很快送达，飞机也可能是工程史上最杰出的产品之一，但这并不意味着它是运送每封信的最佳工具。

如果日常任务中有相当一部分可以在本地或由更小的模型完成，我们就能减轻计算基础设施的压力，把重型资源重新分配给真正需要它们的工作。

但这种可能性不会自动转化为环境收益。廉价人工智能的普及可能大幅增加使用量，而在缺乏协调的情况下运行多个模型，也可能抵消模型缩小带来的部分收益。

因此我认为，下一阶段的竞争不只是减少参数，而是改善任务层经济性（Task-Level Economics）。

系统在得出正确答案前消耗了多少焦耳？结果出现前经过了多少毫秒？有多少任务需要升级到更大的模型？错误或延迟会给用户带来什么影响？

这些指标中蕴藏着广阔的工程空间。

我们也许会发现，最佳系统由三部分构成：处理大多数请求的小模型、负责异常情况的更专业模型，以及只在问题变得复杂时才运行的大模型。

路由器（Router）的工作会像乐团指挥：它并不演奏每一种乐器，却知道每个声部应当何时进入，又应当何时沉默。

一首乐曲不会因为所有乐器都被迫以最大音量演奏而变得更好。

只有当每件乐器都在时间中找到自己的位置，作品才会改善。

到本十年末，我期待什么？

我相信，未来几年将以一种比当前性能比较所暗示的更深刻方式，改变我们与语言模型的关系。

大型模型的竞赛仍会继续，向更复杂的通用能力推进，也仍将是重要的研究领域之一。

但与此同时，一个更安静、也更贴近日常生活的产业会不断壮大：在后台运行的专业模型产业。

我看到三个彼此关联的转变。

第一，选择模型将成为智能本身的一部分。先进系统不只是拥有最强的模型，还要知道如何分配任务、预测难度、估算成本，并为每项任务选择合适的路径。

第二，设备将逐渐从展示智能结果的终端，变成执行相当一部分智能的场所。手机不再只是一块连接远方头脑的屏幕；它会拥有本地能力，并在需要时与更强大的能力协同。

第三，也是我认为最重要的，是我所说的认知适应力（Cognitive Fitness）将会兴起。

这里的“适应力”不意味着一个模型比所有其他模型知道得更多，而是说系统能够迅速适应新任务、增加新专长、用一个组件替换另一个组件，并在不重建整套系统的情况下改进其中一部分。

一家拥有数十款专业模型、覆盖不同行业、方言、文档和声音类型的公司，在满足某些市场需求时，可能比一家只提供单一通用模型的公司更有能力，无论后者多么强大。

原因不是它拥有更多模型，而是问题与合适工具之间的距离更短。

这正是我认为 Misraj 与 Kawn 长期值得关注的地方。

如果团队能把它的研究模型家族转化为一套由中小型模型组成的整合系统，并加入智能路由、持续评估和本地运行能力，那么它就会拥有一种可以通过增加专长来扩展、而不只是把单一模型越做越大的工程方法。

在许多市场中，灵活性可能会比规模更重要。

机构并不总需要一种能够谈论所有事情的智能；它们需要的是一套理解自身流程、在边界内行动、保护数据，并能快速、可靠且以合理成本完成所需工作的系统。

在这样的环境里，拥有一组协调一致的专家，可能比拥有一个百科全书式的演说者更具优势。

从科学层面看，我认为最值得研究人员关注的领域，将是这些头脑之间的边界。

两个模型怎样交换知识而不传递错误？路由器怎样知道一项任务已经超出某个模型的专长？当一个组件发生变化时，如何测试整套系统？小模型能否发展出一种用于协作的中间语言（Inter-Agent Communication），减少每次交接都要重新解释全部上下文的需要？

这些研究问题的吸引力，丝毫不亚于增加模型参数。

它们也可能为未来的沙特与阿拉伯世界发现提供广阔空间。

一个更像蜂巢而非巨人的未来

让我们回到蜂群。

如果只研究一只蜜蜂的大脑，试图以此解释蜂群选择新家的能力，你不会得到完整的故事。

你还必须理解舞蹈、侦察蜂的反应、信息如何累积、停止信号怎样发挥作用，以及一种可能性跨过何种阈值后才会成为集体决定。

蜂群智能不只是所有蜜蜂大脑的总和。

它还来自这些蜜蜂之间的关系如何被组织起来。

我就是这样看待人工智能未来的一个重要部分。

决定性问题未必总是：我们能运行的最大模型是什么？

它也可能是：怎样建立一套系统，让每个组件都知道自己擅长什么、不擅长什么、何时应该工作，以及下一步该把任务交给谁？

2026年8月 Raspberry Pi 的小实验、模型路由研究，以及 Misraj 在语言、语音、文档和检索方面的专业模型家族，都是指向一个更广阔方向的不同信号。

人工智能还没有小到可以在受限设备上解决一切，专业模型也没有消除我们对巨型模型的需要。

但它们已经开始改变我们的观念：智能应该放在哪里？完成有用的工作，究竟需要多少智能？

这个方向最美的地方，或许在于它让更多样的创新形式都能进入未来。

不是每台设备都需要承载一个超级头脑，不是每个模型都需要知道整个世界，也不是每个机构都需要运行自己并不需要的能力。

有时，一台机器只需要深入理解自己的小任务，在身边找到其他同样深入理解各自任务的机器，然后共同协作，又不忘记彼此的边界。

正因如此，我从另一个角度理解 Misraj 的工作：它不是试图把巨人缩小，而是在尝试建立一个由专业头脑构成的生态，让能力像生命系统中的生物一样繁衍并相互协作。

也许终有一天，我们不会知道究竟是哪一个模型修正了文本、理解了文档、帮助孩子阅读，或发现了机器故障。

我们只会注意到，事物运转得更聪明；隐私不再是获得任何帮助都必须支付的代价；小型设备也学会了完成过去看似超出自身能力的事情。

那一天并不会宣告巨型模型的死亡。

它会宣告人工智能的成熟：人工智能不再试图把整个世界塞进一个头脑，而是学会把知识分配给许多头脑。

蜜蜂不需要变得和蜂巢一样大，才能为蜂巢服务。

伟大的智能并不总是知道所有答案；它还在于知道哪个头脑拥有答案、如何找到它，以及何时不去询问任何头脑才是真正的智慧。

参考文献与一手资料

1. [Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms](#) — Science / AAAS.
2. [Stop signals provide cross inhibition in collective decision-making by honeybee swarms](#) — PubMed record — U.S. National Library of Medicine.
3. [Mastering edge AI on Raspberry Pi with LiteRT and Gemma](#) — Raspberry Pi.
4. [Introducing the third generation of Apple Foundation Models](#) — Apple Machine Learning Research.
5. [La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp.; la théorie de la stigmergie](#) — Insectes Sociaux / Springer.
6. [Ant Colony Optimization](#) — About ACO — IRIDIA, Université libre de Bruxelles.
7. [Kuwait 1.5B: An Arabic SLM via Language Injection](#) — arXiv (preprint).

8. [Mutarjim: Advancing Arabic-English Translation with a Small Language Model](#) — arXiv (preprint).
9. [Sadeed: Advancing Arabic Diacritization through Small Language Model Fine-Tuning](#) — arXiv (preprint).
10. [Lahjawi: An Arabic Dialect Translation Suite](#) — Association for Computational Linguistics.
11. [Baseer: A Vision-Language Model for Arabic Document-to-Markdown OCR](#) — arXiv (preprint).
12. [Arabic AI Models: Language, OCR, Speech & Embeddings](#) — Misraj AI / Kawn Lab.
13. [Linguistic Specialization of Arabic in Text Embedding Models via Multi-Teacher Knowledge Distillation](#) — Misraj AI / Kawn Lab.
14. [Nabr — Quranic recitation ASR and Tajweed-aware mispronunciation detection](#) — Misraj AI / Kawn Lab.
15. [Baseer — Arabic document intelligence](#) — Misraj AI / Kawn Lab.
16. [Kawn Embed Light](#) — Misraj AI / Kawn Lab.
17. [Developer Solutions — Seamless API, Kawn Console, and deployment options](#) — Misraj AI.
18. [Workforces](#) — Misraj AI.
19. [FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance](#) — arXiv (preprint).
20. [RouteLLM: Learning to Route LLMs with Preference Data](#) — ICLR 2025.
21. [Mapping the Efficiency Landscape of Small Language Models](#) — IJCAI 2026.
22. [Energy demand from AI — Energy and AI](#) — International Energy Agency.
23. [Executive summary — Key Questions on Energy and AI](#) — International Energy Agency.