



АВТОРСКАЯ АНАЛИТИЧЕСКАЯ СТАТЬЯ

Пчела, остановившая танец

От тайн роевого интеллекта к саудовской лаборатории Kawn: почему будущее, которое не под силу гигантам, могут создать малые специализированные умы?

В 2012 году американский биолог Томас Сили (Thomas D. Seeley) и его коллеги опубликовали в журнале *Science* поразительное открытие о том, как пчелиные рои принимают одно из самых важных для себя решений — выбирают новый дом.

Пчёлы-разведчицы разлетаются в поисках подходящей полости для улья. Найдя многообещающее место, разведчица возвращается к рю и исполняет виляющий танец (Waggle Dance), сообщая сведения о находке. Чем лучше место, тем энергичнее она зовёт других и тем больше пчёл отправляются проверить его самостоятельно.

В улье нет архитектора, который чертит план, нет центральной комиссии, сопоставляющей предложения, и нет матки, отдающей приказ о переезде.

И всё же тысячи пчёл способны прийти к чрезвычайно сложному коллективному выбору.

Но открытие Сили заключалось не только в танце.

Исследователи заметили нечто ещё более удивительное: **некоторые пчёлы не просто отстаивают свой вариант, но и посылают сигналы остановки (Stop Signals) пчёлам, которые продвигают другие места.**

В нейробиологии этот механизм известен как перекрёстное торможение (Cross-Inhibition). Он помогает группе выйти из тупика при выборе между близкими вариантами и не даёт соперничеству продолжаться бесконечно.

Будто природа поняла: интеллекту нужен не только тот, кто скажет «я нашёл решение», но и тот, кто знает, когда сказать «перестань идти этим путём».

Исследователи описали это явление в работе *Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms*.

Это небольшое биологическое наблюдение подводит к вопросу, который гораздо шире жизни пчёл.

Как существа с ограниченными индивидуальными возможностями создают сложное коллективное поведение, если ни один отдельный ум не знает всего?

И не скрывается ли в этом вопросе один из ключей к грядущему будущему искусственного интеллекта?

В последние годы мы привыкли смотреть на развитие языковых моделей через их масштаб: миллиарды параметров (Parameters), всё более широкие контекстные окна (Context Window), способность к рассуждению (Reasoning), стремление к общему искусственному интеллекту (Artificial General Intelligence — AGI), а затем и к искусственному сверхинтеллекту (Artificial Superintelligence — ASI).

Порой эта гонка выглядела как попытка создать сверхсущество, которое прочитает все книги, овладеет всеми языками, разберётся во всех профессиях и ответит на любой вопрос.

Но природа, которая на миллионы лет опередила наши лаборатории в опытах с интеллектуальными системами, предлагает другую философию.

Чтобы группа стала умнее, не обязательно делать умнее каждого её участника.

Иногда гениальность состоит в том, чтобы выстроить отношения между малыми умами, а не бесконечно раздувать один-единственный ум.

Робот, которому не понадобилось облако

11 августа 2026 года Raspberry Pi Foundation опубликовала технический материал, который на первый взгляд выглядел скромнее анонсов гигантских моделей.

Организация показала, как модели семейства Gemma от Google работают в среде выполнения LiteRT на небольшом компьютере Raspberry Pi 5. Среди демонстраций был маленький робот Reachy Mini: он принимает звуковые и зрительные сигналы и отвечает локально, не полагаясь на постоянное соединение с удалённым вычислительным центром.

В опубликованных тестах производительности Gemma 3 270M, содержащая всего 270 миллионов параметров, декодировала текст со скоростью около 22,6 токена в секунду в LiteRT-LM. Файл модели занимал 278 МБ, а пиковое потребление памяти в тесте составляло около 680 МБ.

В самой мультимодальной демонстрации робота использовалась другая модель того же семейства — Gemma 4 E2B, — а обработка звука, изображения и языка была распределена между специализированными компонентами.

Подробности изложены в [официальном техническом материале Raspberry Pi](#).

Модель с 270 миллионами параметров не превзойдёт лучшие универсальные модели в решении научных задач, а маленький робот не сможет соперничать с вычислительными центрами во всех видах работы.

Но главное в этой новости — другое.

Возможности, для которых ещё несколько лет назад требовались удалённые серверы, теперь доступны на компьютере, помещающемся на ладони.

Ещё важнее то, что система не требует от одной модели уметь всё. Один компонент отвечает за зрение, другой — за распознавание речи, языковая модель выбирает ответ, а затем система преобразует текст в речь.

Четыре специальности соединяются и рожают одно действие, которое кажется пользователю естественным.

В июне 2026 года Apple представила третье поколение своих базовых моделей Apple Foundation Models, в том числе модели, работающие непосредственно на устройстве (On-Device Models), среди них AFM 3 Core примерно на три миллиарда параметров, а также модели для собственных серверов компании. Это был не отказ от облака, а продуманное перераспределение работы между устройством и сервером. [Apple Machine Learning Research](#)

Когда я сопоставляю эти два события, я вижу не только успех в уменьшении моделей.

Я вижу начало перемен в самой постановке инженерного вопроса.

Вместо того чтобы спрашивать, как отправить каждую задачу самой большой доступной модели, мы всё чаще будем спрашивать: **какого минимального объёма интеллекта достаточно для этой задачи и где он должен работать?**

Ответ может находиться внутри телефона, промышленного устройства, на сервере организации или в огромной глобальной модели, когда того требует задача.

Суть не в победе одного из этих мест над другими, а в умении разумно выбирать между ними.

Почему муравьиной колонии не нужен генеральный директор?

Задолго до появления искусственного интеллекта животный мир столкнулся с задачей координации огромного числа существ с ограниченными возможностями.

Как муравьиная колония находит дорогу к пище? Как она организует движение тысяч рабочих муравьёв? Как приспосабливается, если путь прерван или появляется новый источник?

В 1959 году французский энтомолог Пьер-Поль Грассе (Pierre-Paul Grassé) ввёл понятие, которое стало важным для изучения сложных систем, — координацию через изменения среды, или стигмергию (Stigmergy).

Смысл в том, что отдельный участник может влиять на поведение других, не посылая им прямых указаний.

Муравей оставляет на пути химический след; другие муравьи реагируют на него, а их движение, в свою очередь, меняет силу маршрутов. Со временем и под воздействием меняющихся условий среды эти следы тоже изменяются.

Из множества локальных взаимодействий возникают упорядоченные структуры.

Ни у одного муравья нет полной карты происходящего в колонии, однако колония способна выполнять работу, которая намного превосходит возможности отдельной особи.

Исследователи называют это коллективным интеллектом (Collective Intelligence), а когда на уровне группы возникают свойства, которых в таком виде нет у её участников, — эмерджентным поведением (Emergent Behavior).

Эти идеи не остались мимолётными литературными метафорами. Они перешли в информатику и роевую робототехнику (Swarm Robotics), помогли разработать методы оптимизации (Optimization), поиска путей и распределения задач.

Один из самых известных примеров — алгоритмы муравьиной колонии (Ant Colony Optimization), разработка которых связана с именем исследователя Марко Дориго (Marco Dorigo).

Я не утверждаю, что нейронные сети должны буквально подражать муравьиным колониям: биологическая метафора — ещё не готовый инженерный проект.

Но глубинный принцип заслуживает внимания:

Большая способность может возникнуть из ограниченных элементов, если способ их сотрудничества устроен лучше.

Именно поэтому меня привлекает один саудовский опыт: мне кажется плодотворнее рассматривать его через призму живых систем, чем помещать в привычную гонку размеров моделей.

Речь о Misraј AI и её исследовательской лаборатории Kawn Lab.

Misraј: когда не требуется ум, знающий всё

В направлении, которое выбрала Misraј, меня интересует не просто наличие малой арабской языковой модели.

Гораздо существеннее то, что компания начала строить **семейство специализированных возможностей**: у каждой своя чёткая функция, область знания, способ обучения и оценки, соответствующий характеру задачи.

В центре этого семейства находится Kuwain 1.5B — малая языковая модель (Small Language Model — SLM) с 1,5 миллиарда параметров, которая стала основой для разработки более специализированных моделей.

Но история семейства становится интереснее, когда мы отходим от универсальной модели и рассматриваем её ветви.

Одна из них — **Mutarjim**, модель для перевода между арабским и английским, созданная на основе Kuwain.

В препринте, размещённом на arXiv в 2025 году, исследователи сообщили, что в отдельных тестах Mutarjim превзошла гораздо более крупные модели, в том числе GPT-4o mini при переводе с английского на арабский по набору Tarjama-25.

Это не утверждение, будто она лучше всех моделей GPT во всех языках и задачах.

Здесь проявляется важный технический парадокс: в пределах строго определённого испытания малая специализированная модель способна обойти универсальную коммерческую модель, которая больше её и сильнее во многих других областях.

Затем идёт **Sadeed**, модель для арабской диакритизации (Arabic Diacritization).

Тому, кто не знаком со сложностью арабского языка, расстановка огласовок может показаться второстепенной задачей.

Но слово «علم» может обозначать знание, флаг или форму глагола «знать». Выбрать верное чтение по одному лишь начертанию букв система не может: ей нужно понимать строй предложения, контекст и связи между словами.

В классических и литературных текстах один небольшой знак способен изменить грамматическую роль слова или смысл фразы.

Поэтому диакритизация — не декоративное упражнение по расстановке огласовок, а задача, в которой сочетаются морфологический анализ (Morphological Analysis), синтаксический анализ (Syntactic Analysis) и разрешение семантической неоднозначности (Semantic Disambiguation).

В препринте о Sadeed команда описала специализированное обучение на тщательно подготовленных арабских данных и представила тестовый набор SadeedDiac-25 для сравнения качества диакритизации текстов разных типов.

Далее следует семейство **Lahjawi** — для перевода между арабскими диалектами и преобразования диалектной речи в современный литературный арабский язык (Modern Standard Arabic — MSA).

Это другая специализация, не та же самая, что перевод между английским и арабским. Диалекты различаются не только лексикой: у них разные конструкции, выражения, социальные оттенки, а порой и значения, которые одно слово приобретает в зависимости от страны и контекста.

Nabr — звуковая модель для чтения Корана: она распознаёт чтение и выявляет некоторые ошибки произношения и таджвида (Tajweed-Aware Mispronunciation Detection).

От неё не требуют быть универсальным помощником, который отвечает на вопросы об истории Римской империи или анализирует бюджет компании.

Её задача — работать в узкоспециализированной звуковой области, где важны долгота гласных, гунна, калькаля, эмфатическое произношение и другие особенности исполнения.

В другой области работает **Baseer** — трёхмиллиардная мультимодальная модель зрения и языка (Vision-Language Model — VLM), предназначенная для понимания арабских документов и преобразования их содержания в структурированный текст с учётом визуального устройства страниц.

Наконец, мы подходим к семейству, в котором я вижу один из главных ключей к будущему, — моделям векторных представлений (Embedding Models).

Наиболее заметные из них — Kawn Embed Light с 300 миллионами параметров, а также Kawn Embed Islamic и Kawn Embed Medical, в каждой из которых по 139 миллионов параметров; эти модели предназначены для поиска знаний в разных областях.

Они не пишут статьи и не выступают чат-ботами. Они превращают тексты в математические представления — векторные вложения (Vector Embeddings), — помогая компьютеру находить фрагменты, наиболее близкие к нужному смыслу, даже если в них использованы другие слова.

По данным MisraJ на авторской странице работы LaTell 2026, её исследователи описали способ специализации арабских моделей векторных представлений с помощью дистилляции знаний от нескольких учителей (Multi-Teacher Knowledge Distillation).

И здесь возникает картина, почти в точности повторяющая идею, с которой началась эта статья.

Перед нами семейство, где ни от одного участника не требуется знать всё.

Одна модель слушает. Другая переводит. Третья расставляет огласовки. Четвёртая понимает отсканированную страницу. Пятая ищет смысл в миллионах документов.

Знания распределены здесь не потому, что команда не сумела создать одну модель, способную на всё, а потому, что сама специализация может быть лучшим инженерным решением при проектировании системы для определённых задач.

На мой взгляд, это одна из самых примечательных идей в работе MisraJ.

Ценность не в том, что каждая малая модель победит всех гигантов, а в том, что целостная система сможет выбирать, кому поручить задачу и когда передать работу дальше.

Когда модели становятся рабочей командой

Представим сцену, близкую к повседневной жизни.

Мать сидит дома. Перед ней школьные бумаги, отчёт с английскими терминами, рядом ребёнок, который хочет услышать арабский текст с огласовками, а ещё — написанное официальным языком школьное сообщение, которое нужно объяснить.

При привычном способе работы она может отправить всё большому языковому помощнику и попросить перевести отчёт, извлечь сведения, расставить огласовки и объяснить указания.

Возможно, помощник справится вполне достойно.

Но можно представить другую архитектуру — такую, где не предполагается, что любая проблема требует вызова одной и той же модели.

Если документ отсканирован как изображение, его текст и структуру извлекает модель понимания документов, например Baseer.

Если текст нужно перевести, задача переходит к Mutarjim.

Если для чтения арабского текста нужны огласовки, вызывается Sadeed.

Если вопрос касается смысла фразы в определённых документах, модель векторных представлений, например Kawn Embed Light, сначала находит подходящие фрагменты, а затем соответствующий языковой компонент формулирует на их основе ответ.

Не обязательно запускать все эти модели каждый раз. Для одних запросов достаточно одной, для других нужна короткая цепочка, а третьи требуют параллельной обработки.

В зависимости от устройства такая система может использовать маршрутизацию моделей (Model Routing) или каскад моделей (Model Cascading); если к ним добавляются принятие решений, применение инструментов и управление работой, мы входим в область мультиагентных систем (Multi-Agent Systems).

Это не то же самое, что архитектура смеси экспертов (Mixture of Experts — MoE) внутри одной нейронной сети. Там задействуются отдельные части одной модели; здесь речь идёт о координации независимых моделей с разными функциями.

Для инженера различие важно, но саму идею понять нетрудно.

Вы не зовёте офтальмолога чинить кондиционер и не просите профессионального переводчика заменить дипломированного бухгалтера только потому, что оба получили образование.

Почему же тогда мы порой настаиваем на том, чтобы поручать всё одной языковой модели?

В 2023 году исследователи Стэнфорда, в том числе Линцзяо Чэнь, Матей Захария и Джеймс Цзоу, опубликовали препринт *FrugalGPT*. В нём изучалось, как направлять запросы разным моделям, чтобы снижать стоимость инференса, сохраняя качество в пределах заданных тестов.

Затем в работе *RouteLLM*, опубликованной на ICLR 2025, исследовалась обучаемая маршрутизация моделей на основе данных о предпочтениях.

Мне нравится в этих исследованиях то, что часть интеллекта они переносят на уровень выше самой модели.

Вместо того чтобы добавлять новые знания в каждую модель, мы можем улучшить систему, научившись определять, **какую модель вообще следует использовать**.

С этой переменной умение распределять задачи между моделями может стать с инженерной точки зрения не менее ценным, чем умение разрабатывать одну из них.

Возможно, некоторые задачи, для которых, как нам казалось, нужен ум мощнее, на самом деле нуждались в более разумном распределении работы.

Подход	Где принимается решение?	Что взаимодействует?
Маршрутизация моделей (Model Routing)	Между независимыми моделями, в зависимости от задачи	Отдельные специализации
Каскад моделей (Model Cascading)	На последовательных этапах, по мере необходимости	Модели, которым задача передаётся по очереди
Смесь экспертов (Mixture of Experts)	Внутри архитектуры одной модели	Выбранные части нейронной сети

Концептуальное сопоставление по материалу этого раздела; это не сравнение производительности.

Величайшее умение агента — знать, когда остановиться

Здесь я возвращаюсь к пчеле Томаса Сили.

Слаженность решения роя обеспечивается не только способностью пчёл предлагать новые места, но и механизмами, которые не дают слабым вариантам задерживаться и помогают группе прийти к выбору.

Это возвращает меня к мысли, которую упускают многие сторонники ИИ-агентов.

Если распределить задачу между десятью слабыми моделями, система не обязательно станет сильной лишь потому, что участников прибавилось.

Ошибки могут переходить от модели к модели: первая начинает с неверного толкования, вторая строит на нём свои выводы, третья уверенно пересказывает их — и в результате ответ звучит чрезвычайно убедительно, хотя был ошибочен с первого шага.

Поэтому инженерия сотрудничества не заканчивается выбором специалиста.

Ей необходимы оценка неопределённости (Uncertainty Estimation), калибровка уверенности (Confidence Calibration), обнаружение данных вне распределения (Out-of-Distribution Detection) и проверка результатов (Output Verification).

Если малая модель не достигает требуемого уровня уверенности, система должна уметь передать задачу более крупной модели или человеку на проверку.

На инженерном языке система должна уметь воздержаться от ответа: нельзя превращать ответ в священный долг, который необходимо исполнить при любой степени сомнения.

В медицине, финансах и праве это свойство может оказаться важнее красноречия или скорости генерации текста.

Модель, которая понимает, что документ неполон или найденных сведений недостаточно, полезнее той, что создаёт безупречно гладкий ответ из ничего.

Я ожидаю, что следующие поколения специализированных агентов будут соперничать в качестве этого решения не меньше, чем в качестве самих ответов.

Вопрос будет звучать не просто так: может ли агент выполнить задачу?

Он будет звучать иначе: понял ли агент, какую модель вызвать, что проверить, когда попросить помощи и когда остановиться?

Интеллект здесь становится больше, чем способностью говорить: он превращается в способность управлять знанием и его пределами.

Иногда самый умный участник команды — тот, кто не даёт ей совершить следующую ошибку.

Самый быстрый ум — не всегда самый большой

Обсуждая эффективность малых моделей, легко впасть в другое упрощение: решить, будто меньшее число параметров непременно делает модель быстрее и снижает её энергопотребление.

Но инженерия вычислительных систем сложнее.

Производительность зависит от архитектуры модели, способа представления весов, эффективности среды инференса, объёма входных и выходных данных, скорости памяти, типа процессора, а также затрат на загрузку моделей и их хранение в памяти.

Квантизация (Quantization) позволяет уменьшить число битов, которыми представлены веса, — например, перейти от 16-битного формата FP16 к четырёхбитному INT4 и тем самым значительно сократить необходимый для весов объём памяти.

Теоретически модели со 139 миллионами параметров при четырёхбитном представлении требуется около 70 МБ только для хранения весов — без учёта метаданных квантизации, временной памяти и накладных расходов среды выполнения.

Модели с 1,5 миллиарда параметров при той же точности может потребоваться около 750 МБ только для весов, ещё до учёта остальных требований к памяти.

Поэтому разумно рассматривать возможность запуска некоторых классов таких моделей на телефонах и периферийных устройствах, но это не означает, что каждая модель в каталоге MisraJ была испытана и показана на конкретном телефоне или Raspberry Pi.

Возможность загрузить модель ещё не означает, что её удастся запускать с приемлемой скоростью.

Специализация может также снизить потребность модели в длинных ответах или обработке огромного контекста, уменьшая стоимость выполнения всей задачи даже без существенного сокращения размера самой модели.

В исследовании *Mapping the Efficiency Landscape of Small Language Models*, опубликованном на IJCAI 2026, были рассмотрены более семидесяти малых моделей. Авторы показали, что самая маленькая модель не всегда оказывается самой эффективной, а соотношение энергозатрат и производительности устроено сложнее, чем рейтинг по числу параметров.

Это важный результат.

Разумная метрика — не только токены в секунду и не только джоули на токен.

Ближе к жизни людей и организаций другой показатель: **сколько времени и ресурсов требуется системе, чтобы правильно выполнить задачу.**

Малая модель может медленно генерировать текст в свободной форме, но быть исключительно эффективной при классификации короткой фразы. Более крупной модели может понадобиться меньше шагов для сложной задачи, и потому экономически она окажется лучшим выбором.

Инженерная изобретательность не в том, чтобы всегда использовать самую маленькую модель.

Она в том, чтобы выбрать модель, которая даёт требуемый результат при наименьшей общей стоимости — в заданных пределах точности, времени и приватности.

Возможно, мы заново открываем старый природный принцип: колония не посылает всех своих обитателей к одному цветку и не выделяет одинаковые ресурсы на каждую задачу.

День, когда телефон станет местом для интеллекта, а не просто окном к нему

Я думаю, одна из важных социальных перемен ближайших лет произойдёт не на экранах больших технологических презентаций, а в привычных повседневных вещах.

Телефон, автомобиль, домашний медицинский прибор, школьный компьютер, заводская камера — все они могут обрести специализированные интеллектуальные возможности, работающие рядом с нужными им данными.

Устройству не нужно знать весь мир.

Телефону матери может понадобиться одна модель для локального распознавания арабской речи, другая — чтобы помочь ребёнку прочитать текст, третья — чтобы упорядочить документы, и при этом ни одна подробность её жизни не отправится на

удалённый сервер.

Заводу может понадобиться модель для извлечения сведений из отчётов о техническом обслуживании, другая — для классификации предупреждений, третья — для поиска эксплуатационных инструкций; данные об оборудовании при этом останутся внутри заводской сети благодаря локальному развёртыванию (On-Premises Deployment).

В школе одна специализированная модель может исправлять чтение, другая — упрощать тексты, третья — отслеживать успехи ученика в рамках ясной политики защиты его данных.

Это не призыв заменить учителя, врача или инженера.

Это попытка сделать менее обременительными мелкие дела, которые отнимают их время, и обеспечить некоторые возможности на месте, даже когда связь слаба или отсутствует.

Одним из важнейших возможных следствий может стать превращение приватности из дополнительной услуги в свойство самой архитектуры.

Если данные изначально не покидают устройство, существенная часть рисков, связанных с их передачей и хранением вне него, уменьшается. Однако само устройство, локальные модели и журналы их использования по-прежнему нуждаются в защите.

В организациях локальный инференс (Local Inference) и развёртывание в изолированных сетях (Air-Gapped Deployment) открывают путь к приложениям, которые трудно предоставлять через общедоступные облачные сервисы.

Здесь я вижу важное преимущество семейств специализированных арабских моделей.

Вместо того чтобы обращаться с каждым запросом к огромной универсальной модели, можно создать систему, в которой малые модели работают внутри периметра организации, сложные задачи остаются на более крупных внутренних серверах, а то, что разрешает политика, при необходимости передаётся внешним сервисам.

Misraj уже предлагает продукты для разработки на основе своих моделей, например Seamless API для построения цепочек обработки и маршрутизации и Workforces для операционных агентов, а также варианты локального и корпоративного развёртывания.

Это архитектурные элементы, из которых можно создавать многомодельные системы. Реальное преимущество каждой конфигурации перед альтернативами зависит от задачи, оборудования и испытаний в эксплуатации.

Возможно, самое привлекательное человеческое следствие состоит в том, что пользователю всё реже придётся знать имя искусственного интеллекта.

Некоторые задачи растворятся внутри телефона или приложения: фраза будет переведена там, где нужен перевод, документ — понят там, где требуется понимание, а текст — исправлен там, где возникает ошибка.

Признаком зрелости искусственного интеллекта может стать момент, когда мы перестанем замечать, что пользуемся им всякий раз, когда он нам помогает.

Так же как мы не упоминаем электричество каждый раз, когда зажигаем свет в комнате.

Когда арабский становится инженерным преимуществом

Есть и другая сторона малых специализированных моделей, выходящая за рамки памяти и энергии.

Трудности арабского языка — его морфологическое, диалектное и смысловое разнообразие — распределены между задачами неравномерно.

Задача системы поиска по арабским медицинским документам отличается от распознавания ошибок в чтении Корана, от перевода между диалектами стран Персидского залива и Магриба и от извлечения таблиц из отсканированного документа.

Универсальная модель должна заключить широкий спектр этих задач в одном наборе возможностей.

Специализация позволяет направить данные, обучение и оценку на конкретную проблему.

Возьмём Kawn Embed Medical.

В медицинских документах пациент может описать симптом разговорными словами, тогда как соответствующее понятие в клинической записи выражено профессиональным термином. Задача специализированной модели векторных представлений — сблизить разные формулировки одного смысла и тем самым помочь найти нужную информацию.

Обнаружив подходящий текст, модель не ставит диагноз; она выполняет точную поисковую функцию, необходимую медицинским приложениям.

Для Kawn Embed Islamic задача иная: связывать запросы с подходящими текстами Корана, хадисов, тафсира и фикха, сохраняя важность ссылки на источник.

Две модели могут быть одинаковы по размеру, но различаться обучающими данными, семантическим пространством и критериями успеха.

В этой гибкости я вижу широкие возможности для развития того, что можно назвать **гиперлокальным специализированным ИИ**.

Такие модели понимают диалекты, профессиональную лексику и документы, созданные определёнными сообществами, но им не приходится заново осваивать все знания мира.

В своих областях они могут развиваться быстрее универсальных моделей, поскольку обновить данные для узкой задачи и оценить последствия порой проще, чем изменить всеобъемлющую систему знаний.

Это не значит, что гигант стал бесполезен.

Продвинутые универсальные модели сохраняют значение для сложных рассуждений, незнакомых ситуаций, соединения разных областей знания, а также в роли контролёров или резервных исполнителей для трудных случаев.

Но во многих системах они могут стать **возможностью, к которой обращаются при необходимости, а не возможностью, которую автоматически запускают ради самых простых дел**.

Это серьёзная перемена в философии проектирования.

Экономика, которая платит за гениальность лишь при необходимости

В отчёте об энергетике и искусственном интеллекте Международное энергетическое агентство оценило мировое потребление электроэнергии центрами обработки данных примерно в 415 тераватт-часов в 2024 году; в более позднем обновлении агентство спрогнозировало, что к 2030 году оно приблизится к 950 тераватт-часам.

Искусственный интеллект — не единственная причина энергопотребления центров обработки данных, но он стал одним из главных факторов роста спроса.

По мере того как модели проникают во все виды деятельности, возникает новый вопрос.

Если миллионы людей в течение дня используют ИИ для мелких задач, разумно ли с вычислительной точки зрения отправлять каждую из них самым мощным и ресурсоёмким моделям?

Это всё равно что перевозить бумажное письмо грузовым самолётом.

Письмо может прибыть быстро, а самолёт — быть одним из лучших инженерных достижений, но от этого он не становится лучшим средством доставки каждого письма.

Если значительную часть повседневных задач удастся выполнять локально или с помощью меньших моделей, мы сможем снизить нагрузку на вычислительную инфраструктуру и направить тяжёлые ресурсы на работу, которая действительно их заслуживает.

Однако сама по себе такая возможность не гарантирует экологической экономии. Дешёвый ИИ может привести к гораздо более широкому использованию, а одновременный запуск нескольких моделей без координации способен растратить часть выигрыша от их уменьшения.

Поэтому я полагаю, что следующим полем конкуренции станет не просто сокращение числа параметров, а улучшение **экономики на уровне задачи (Task-Level Economics)**.

Сколько джоулей израсходовала система, прежде чем пришла к правильному ответу? Сколько миллисекунд прошло до появления результата? Какую долю задач пришлось передать более крупной модели? Как ошибка или задержка повлияла на пользователя?

Эти показатели открывают широкое пространство для инженерных решений.

Возможно, лучшей окажется система, в которой малая модель обрабатывает большинство запросов, более специализированная берётся за необычные случаи, а большая запускается только тогда, когда задача становится сложной.

Маршрутизатор будет похож на дирижёра оркестра: он не играет на каждом инструменте, но знает, когда должен вступить каждый голос и когда ему следует молчать.

Музыкальное произведение не становится лучше, если заставить все инструменты играть на предельной громкости.

Оно становится лучше, когда каждому инструменту отведено своё время.

Чего я ожидаю к концу этого десятилетия?

Я думаю, что в ближайшие годы наши отношения с языковыми моделями изменятся глубже, чем можно предположить по нынешним сравнениям производительности.

Гонка больших моделей продолжится, а движение к более сложным универсальным возможностям останется одним из важнейших направлений исследований.

Но рядом вырастет более тихая и близкая к повседневности отрасль — отрасль специализированных моделей, работающих в фоновом режиме.

Я вижу впереди три взаимосвязанные перемены.

Во-первых, **выбор модели станет частью самого интеллекта**. У продвинутой системы будет не просто самая сильная модель: она будет уметь распределять задачи, предсказывать их сложность, оценивать стоимость и выбирать подходящий путь.

Во-вторых, устройства постепенно перейдут от показа результатов интеллекта к выполнению значительной части его работы. Телефон станет не просто экраном, соединённым с удалённым умом: у него появятся локальные возможности, которые при необходимости объединяются с более крупными.

В-третьих — и для меня это важнее всего — распространится то, что я называю **когнитивной приспособляемостью (Cognitive Fitness)**.

Под приспособляемостью здесь понимается не превосходство модели в объёме знаний. Речь о способности системы быстро освоить новую задачу, добавить специализацию, заменить один компонент другим и улучшить одну часть, не перестраивая всё целиком.

Компания с десятками специализированных моделей для разных отраслей, диалектов, документов и типов речи может лучше удовлетворять некоторые потребности рынка, чем компания, предлагающая лишь одну универсальную модель, какой бы мощной та ни была.

Не потому, что у неё больше моделей, а потому, что расстояние между проблемой и подходящим инструментом короче.

Именно поэтому, на мой взгляд, за Misraj и Kawn стоит наблюдать в долгосрочной перспективе.

Если команда сумеет превратить своё исследовательское семейство в целостную систему малых и средних моделей с умной маршрутизацией, постоянной оценкой и возможностью локального развёртывания, у неё появится инженерный подход,

который масштабируется добавлением специализаций, а не только увеличением одной модели.

На многих рынках гибкость может стать важнее размера.

Организациям не всегда нужен интеллект, способный обсуждать всё на свете. Им нужна система, которая понимает их работу, действует в установленных границах, защищает данные и выполняет необходимые задачи быстро, надёжно и за разумную цену.

В такой среде скоординированная группа специалистов может оказаться более сильным преимуществом, чем один энциклопедически образованный оратор.

На научном уровне, как мне кажется, важнейшей областью для исследователей станет изучение границ между этими умами.

Как двум моделям обмениваться знаниями, не передавая друг другу ошибки? Как маршрутизатор определит, что задача вышла за пределы специализации модели? Как испытывать систему целиком, когда один из её компонентов меняется? Могут ли малые модели выработать промежуточный язык сотрудничества, который избавит от необходимости заново объяснять весь контекст при каждой передаче задачи?

Эти исследовательские вопросы не менее увлекательны, чем увеличение числа параметров.

Они могут стать плодородной почвой для будущих саудовских и арабских открытий.

Будущее, больше похожее на улей, чем на гиганта

Я возвращаюсь к пчелиному рою.

Если попытаться объяснить его способность выбирать новый дом, изучая только мозг одной пчелы, всей истории мы не увидим.

Нужно понять танец, ответы разведчиц, накопление информации, сигналы остановки и тот порог, за которым возможность превращается в коллективное решение.

Интеллект роя — не просто сумма пчелиных мозгов.

Это ещё и способ организации отношений между ними.

Так я вижу важную часть будущего искусственного интеллекта.

Решающим вопросом не всегда будет: какую самую большую модель мы можем запустить?

Возможно, он будет звучать так: как построить систему, где каждый компонент знает, что он делает хорошо, чего не умеет, когда ему следует вступить в работу и кому передать задачу дальше?

Небольшой эксперимент Raspberry Pi в августе 2026 года, исследования маршрутизации моделей и семейство MisraJ, специализированное на языке, речи, документах и поиске, — разные признаки одного более широкого направления.

Искусственный интеллект ещё не стал настолько мал, чтобы решать на ограниченном устройстве все задачи, а специализированные модели не отменили потребность в гигантах.

Но они уже начали менять наше представление о том, где должен находиться интеллект и сколько его нужно для полезной работы.

Самая прекрасная сторона этого направления, возможно, в том, что оно открывает будущее для более разнообразных форм инноваций.

Не каждому устройству нужен сверхразум, не каждой модели нужно знать весь мир и не каждой организации нужно запускать возможности, в которых она не нуждается.

Порой достаточно, чтобы машина глубоко понимала свою небольшую задачу, находила рядом другие машины, столь же глубоко понимающие свои задачи, и чтобы все они сотрудничали, не забывая о собственных пределах.

Именно поэтому я смотрю на работу MisraJ под другим углом: не как на попытку уменьшить гиганта, а как на попытку построить **экологию специализированных умов**, где способности могут множиться и сотрудничать, как живые существа в экосистеме.

Возможно, однажды мы не будем знать название модели, которая исправила текст, поняла документ, помогла ребёнку читать или обнаружила неисправность в машине.

Мы лишь заметим, что всё работает разумнее, что приватность больше не служит обязательной платой за любую помощь, а малые устройства научились делать то, что прежде казалось им не по силам.

Этот день не возвестит о смерти гигантских моделей.

Он возвестит о зрелости искусственного интеллекта — о времени, когда тот перестанет пытаться вместить весь мир в один ум и научится распределять знания между множеством умов.

Пчеле не нужно становиться величиной с улей, чтобы служить ему.

Великий интеллект — не всегда знание всех ответов; это знание того, у какого ума есть ответ, как к нему обратиться и когда мудрость состоит в том, чтобы не спрашивать никого.

Источники и первичные материалы

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.