



OPINION AND ANALYSIS

The Bee That Stopped the Dance

From the secrets of swarm intelligence to Saudi Arabia's Kawn Lab: why small specialized minds may build a future that giants cannot

In 2012, the American biologist Thomas D. Seeley and his colleagues published a striking discovery in *Science* about how honeybee swarms make one of their most consequential decisions: choosing a new home.

Scout bees leave to search for a suitable cavity. When one finds a promising site, it returns to the swarm and performs a waggle dance, conveying information about the location. The better the site, the more vigorous the recruitment, and the more bees leave to inspect it for themselves.

There is no architect in the hive drawing the plan, no central committee comparing proposals, and no queen issuing an order to move.

Yet thousands of bees can arrive at a remarkably complex collective choice.

But Seeley's discovery was not only about the dance.

The researchers observed something more surprising: **some bees do not merely advocate for their own option; they send stop signals to bees promoting competing sites.**

In neuroscience, this mechanism is known as cross-inhibition. It helps a group break a deadlock between closely matched alternatives and prevents competition from continuing indefinitely.

It is as if nature understood that intelligence needs not only someone to say, “I found the answer,” but someone who knows when to say, “Stop following this path.”

The researchers documented the phenomenon in *Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms*.

This small biological observation leads to a question much larger than the life of bees.

How can creatures with limited individual abilities produce advanced collective behavior when no individual mind knows everything?

And might that question hold one of the keys to the coming future of artificial intelligence?

In recent years, we have become accustomed to viewing the development of language models through their scale: billions of parameters, wider context windows, reasoning ability, the ambition of artificial general intelligence (AGI), and then artificial superintelligence (ASI).

At times, the race has looked like an attempt to build a super-creature that reads every book, masters every language, understands every profession, and answers every question.

But nature, which has had millions of years more than our laboratories to experiment with intelligent systems, offers another philosophy.

Not every individual has to become smarter for the group to become smarter.

Sometimes genius lies in engineering the relationships among small minds, rather than inflating one mind without end.

The robot that did not need the cloud

On 11 August 2026, the Raspberry Pi Foundation published a technical account that initially seemed less dramatic than announcements of giant models.

It demonstrated models from Google’s Gemma family running through the LiteRT runtime on the Raspberry Pi 5, including an experiment with a small robot called Reachy Mini that can receive audio and visual signals and respond locally, without depending on a permanent connection to a distant computing centre.

In the published performance tests, Gemma 3 270M, with only 270 million parameters, decoded text at about 22.6 tokens per second using LiteRT-LM, with a 278 MB model file and peak memory use of about 680 MB in the test.

The multimodal robot experiment itself used another model in the family, Gemma 4 E2B, within a system that distributes audio, image, and language processing across specialized components.

The details appear in [Raspberry Pi's official technical experiment](#).

A 270-million-parameter model will not surpass the greatest general models at solving scientific puzzles, and the little robot will not compete with data centres at every task.

But the important news lies elsewhere.

Capabilities that required remote servers only a few years ago can now run on a computer held in the palm of a hand.

More importantly, the system does not ask one model to do everything. There is a component for vision, another for speech recognition, a language model to determine the response, and then a text-to-speech system.

Four specialties meet to produce one behavior that feels natural to the user.

In June 2026, Apple announced the third generation of its Apple Foundation Models, including on-device models such as AFM 3 Core at roughly three billion parameters, alongside models for its private servers. This was not an abandonment of the cloud, but a considered redistribution of work between device and server. [Apple Machine Learning Research](#)

When I place these two developments together, I do not see only success in shrinking models.

I see the beginning of a shift in the engineering question itself.

Rather than asking how to send every task to the largest model available, we will increasingly ask: **what is the smallest amount of intelligence sufficient for this task, and where should it run?**

The answer may be inside the phone, in an industrial device, on an institutional server, or in a large global model when the need warrants it.

The goal is not for one of these locations to defeat the others, but to choose intelligently among them.

Why does an ant colony not need a CEO?

Long before artificial intelligence, the animal world confronted the problem of coordinating vast numbers of creatures with limited capacities.

How does an ant colony find food? How does it organize the movement of thousands of workers? How does it adapt when a path is cut off or a new source appears?

In 1959, the French entomologist Pierre-Paul Grassé introduced a concept that became important to the study of complex systems: stigmergy.

The idea is that an individual can affect the behavior of others without sending them direct instructions.

An ant leaves a chemical trace along a path; other ants respond to it, and their movement in turn changes the strength of the routes. These traces are affected by time and changing environmental conditions.

Orderly patterns emerge from many local interactions.

No single ant has a complete map of what is happening in the colony, but the colony can accomplish work far beyond what the individual can do.

Researchers call this collective intelligence, or emergent behavior when properties appear at the group level that do not exist in the same form in its members.

These ideas were not passing literary metaphors. They moved into computer science and swarm robotics, contributing to methods for optimization, path finding, and task distribution.

Among the best known are ant colony optimization algorithms, whose development is associated with the researcher Marco Dorigo.

I do not claim that neural networks should literally imitate ant colonies; a biological metaphor is not a ready-made engineering design.

But the deeper principle deserves attention:

Great capability can arise from limited units when their method of cooperation is superior.

This is precisely why a Saudi experiment catches my attention—one I find more fruitful to understand through the lens of living systems than through a conventional race over model size.

It is the work of Misraj AI and its research laboratory, Kawn Lab.

Misraj: when the requirement is not one mind that knows everything

What interests me about Misraj's direction goes beyond its having a small Arabic language model.

More consequentially, it has begun to build **a family of specialized capabilities**, each with a clear function, a field of knowledge, and a method of training and evaluation suited to the task.

At the centre of this family is Kuwain 1.5B, a small language model with 1.5 billion parameters, used principally to develop more specialized models.

But the family becomes more interesting when we step away from the general model and look at its branches.

There is **Mutarjim**, specialized in Arabic–English translation and developed on Kuwain.

In their [2025 arXiv preprint](#), the researchers reported results in which Mutarjim surpassed much larger models on specific tests, including GPT-4o mini for English-to-Arabic translation on the Tarjama-25 benchmark.

This is not a claim that it is better than every GPT model in every language and task.

It presents a technically important paradox: within a defined testing domain, a small specialized model can exceed a general commercial model that is larger and more capable in many other domains.

Then there is **Sadeed**, specialized in Arabic diacritization.

Diacritization may seem marginal to anyone unfamiliar with Arabic's complexity.

But a word such as “علم” may mean knowledge, a flag, or the verb “he knew.” A system cannot choose the correct reading from the shape of the letters alone. It must understand sentence structure, context, and relations among words.

In classical and literary texts, a small vowel mark can change a word's grammatical function or a phrase's meaning.

Diacritization is therefore not a decorative exercise in adding vowels; it combines morphological analysis, syntactic analysis, and semantic disambiguation.

In the [Sadeed preprint](#), the team used specialized training on carefully curated Arabic data and introduced the SadeedDiac-25 evaluation benchmark across different text types.

Then comes the **Lahjawi** family, for translation across Arabic dialects and conversion into Modern Standard Arabic (MSA).

This specialty differs from English–Arabic translation. Dialects vary not only in vocabulary but in structures, expressions, social connotations, and sometimes the meaning a word carries in a particular country and context.

Nabr is a speech model specialized in Qur'an recitation, recognizing recitation and detecting some pronunciation and tajweed errors.

It is not asked to be a general assistant that answers questions about the history of the Roman Empire or analyzes a company budget.

Its task is to approach a highly specialized vocal field involving elongation, nasalization, qalqalah, emphatic articulation, and performance characteristics.

In another field, **Baseer** is a three-billion-parameter vision-language model specialized in understanding Arabic documents and converting their contents into structured text while accounting for the visual structure of pages.

Then we reach a family I see as holding one of the most important keys to the future: embedding models.

The most prominent are Kawn Embed Light, at 300 million parameters, and Kawn Embed Islamic and Kawn Embed Medical, at 139 million parameters each, specialized in knowledge retrieval across different domains.

These models do not act as article writers or chatbots. They turn texts into mathematical representations, vector embeddings, that help a computer find passages closest to the intended meaning even when the vocabulary differs.

According to Misraj, its researchers presented work at LaTell 2026 describing a way to specialize Arabic embedding models through multi-teacher knowledge distillation.

Here, the picture nearly matches the idea with which this essay began.

We have a family in which no member is required to know everything.

One model listens. Another translates. A third adds diacritics. A fourth understands the scanned page. A fifth searches for meaning across millions of documents.

Knowledge was not distributed here because the team failed to make one model do everything, but because specialization itself may be the best engineering decision when designing a system for defined tasks.

In my view, this is one of the ideas in Misraj's work most deserving of attention.

The value lies not in every small model defeating every giant, but in an integrated system choosing which model takes a task and when the work should move to another.

When models become a working team

Let us imagine a scene close to daily life.

A mother sits at home with school papers, a report containing English terminology, a child who wants to hear an Arabic text with diacritics, and a formally written school message that needs explanation.

In a conventional usage model, she can send everything to a large language assistant and ask it to translate the report, extract information, add diacritics, and explain the instructions.

The assistant may perform the task competently.

But we can imagine another design, one that does not assume every problem requires calling the same model.

If the paper is a scanned image, a document-understanding model such as Baseer extracts its text and structure.

If the text needs translation, the task moves to Mutarjim.

If Arabic text needs diacritics for reading, Sadeed is called.

If the question concerns the meaning of a phrase appearing in specific documents, an embedding model such as Kawn Embed Light searches for relevant passages before an appropriate language component frames an answer from them.

It is not necessary to run all these models every time. Some requests need one model, some a short chain, and some parallel processing.

Depending on its design, this structure may involve model routing or model cascading; it enters the domain of multi-agent systems when decision-making, tool use, and work management are added.

It is not the same as the mixture of experts (MoE) architecture within a single neural network. That design activates selected parts of the same model; here we are discussing coordination among independent models with different functions.

The distinction matters to the engineer, but the idea is easy to understand.

You do not call an ophthalmologist to repair an air conditioner, or ask a professional translator to substitute for a chartered accountant simply because both are educated.

Why, then, do we sometimes insist on assigning everything to one language model?

In 2023, Stanford researchers including Lingjiao Chen, Matei Zaharia, and James Zou released the FrugalGPT preprint, which studied how requests can be routed among different models to reduce inference cost while preserving performance quality in defined tests.

Then RouteLLM, published at ICLR 2025, studied learned model routing from preference data.

What I like about this research is that it moves part of intelligence to a level above the model itself.

Rather than adding new knowledge to every model, we can improve the system by learning **which model we should use in the first place**.

In this shift, the skill of distributing tasks among models may become as valuable in engineering terms as the skill of developing one of them.

Perhaps some problems we thought needed a greater mind really needed better distribution of work.

Design	Where is the choice made?	What works together?
Model routing	Between independent models, according to the task	Separate specialisms
Model cascading	Across successive stages as needed	Models that receive the task in sequence
Mixture of experts	Inside a single model architecture	Selected parts of the network

A conceptual comparison drawn from this section, not a performance benchmark.

The greatest skill an agent may possess: knowing when to stop

Here I return to Thomas Seeley’s bee.

What makes the swarm’s decision coherent is not only the bees’ ability to propose new sites, but the presence of mechanisms that prevent weak options from persisting and help the group reach a decision.

This brings me back to a point many enthusiasts of AI agents overlook.

If you distribute a task among ten weak models, the system does not necessarily become strong merely because the number of participants has increased.

Errors may pass from model to model: one begins with a mistaken interpretation, the second builds on it, and the third restates it confidently, until the final answer is highly persuasive yet wrong from its first step.

The engineering of cooperation therefore does not end with selecting a specialist.

It needs uncertainty estimation, confidence calibration, out-of-distribution detection, and output verification.

When a small model cannot reach the required confidence level, the system should know how to escalate the task to a larger model or to human review.

In engineering terms, the system must be able to abstain; it must not treat answering as a sacred duty regardless of the degree of doubt.

In health, finance, and law, this property may matter more than eloquence or speed of text generation.

A model that knows a document is incomplete or that retrieved information is insufficient is more useful than one that produces a polished answer from nothing.

I expect coming generations of specialized agents to compete over the quality of this decision as much as they compete over the quality of answers.

Not simply: can the agent complete the task?

But: did it know which model to call, what to verify, when to ask for help, and when to stop?

Here, intelligence becomes more than the capacity to speak; it becomes the capacity to manage knowledge and its limits.

Sometimes the smartest member of a team is the one who prevents it from making the next mistake.

The fastest mind is not always the largest

When we discuss the efficiency of small models, it is easy to fall into another simplification: believing that fewer parameters necessarily make a model faster and less energy-intensive.

But the engineering of computing systems is more complex than that.

Performance depends on model architecture, how weights are represented, the efficiency of the inference runtime, input and output size, memory speed, processor type, and the cost of loading models and keeping them in memory.

Through quantization, the number of bits used to represent weights can be reduced—for example, from 16-bit FP16 to four-bit INT4—greatly reducing the storage required for weights.

In theory, a 139-million-parameter model needs roughly 70 MB to store its weights at four bits, before quantization metadata, temporary memory, and runtime overhead are counted.

A 1.5-billion-parameter model may need roughly 750 MB for weights alone at the same precision, before its other memory requirements are added.

It is therefore reasonable to consider running classes of these models on phones and edge devices, but that does not mean every model in Misraj's catalogue has been tested and demonstrated on a particular phone or on a Raspberry Pi.

Being able to load a model is not the same as being able to run it at a suitable speed.

Specialization may also reduce a model's need to produce long answers or process enormous contexts, improving the cost of the complete task even without a major change in model size.

In a study published at IJCAI 2026, *Mapping the Efficiency Landscape of Small Language Models* examined more than seventy small models and found that the smallest is not necessarily the most efficient in every case, and that the relationship between energy and performance is more complex than ranking models by parameter count.

This result matters.

A smarter metric is not tokens per second alone, nor joules per token.

The measure closer to the lives of people and institutions is **how long a system takes and how many resources it consumes to complete a task correctly.**

A small model may be slow at free-form generation yet exceptionally efficient at classifying a short sentence. A larger model may need fewer steps for a complex problem and therefore be the better economic choice.

Engineering genius does not lie in always using the smallest model.

It lies in using the model that achieves the required result at the lowest total cost, within the limits of accuracy, time, and privacy.

We are perhaps rediscovering an old principle in nature: a colony does not send every member to one flower, nor allocate the same resources to every task.

The day the phone becomes a place for intelligence, not merely a window onto it

I believe one important social shift in the coming years will not happen on the screens of major technology presentations, but in familiar daily objects.

Phones, cars, home medical devices, school computers, and factory cameras are all candidates to gain specialized intelligence that runs close to the data it needs.

The device does not need to know the whole world.

A mother's phone may need one model that recognizes Arabic speech locally, another that helps her child read a text, and a third that organizes her documents, without sending every detail of her life to a distant server.

A factory may need one model to extract information from maintenance reports, another to classify alerts, and a third to retrieve operating instructions while equipment data remains within the factory network through on-premises deployment.

At a school, one specialized model may correct reading, another simplify texts, and a third track a student's progress under a clear policy for protecting the student's data.

This is not a call to replace the teacher, doctor, or engineer.

It is an attempt to make the small tasks that consume their time less burdensome and to make some capabilities available locally even when the connection is weak or interrupted.

One important possible consequence is that privacy becomes a design property rather than an added service.

If data never leaves the device in the first place, a significant share of the risks attached to moving and storing it elsewhere diminishes, while the device itself, local models, and their usage logs still need protection.

Within institutions, local inference and air-gapped deployment make possible applications that are difficult to provide through public cloud services.

Here I see an important advantage for families of specialized Arabic models.

Rather than relying on a huge general model for every request, a system can run small models within institutional boundaries, keep difficult work on larger internal servers, or escalate what policy permits to outside services when necessary.

Misraj already presents products for building on its models, such as Seamless API for composing processing and routing paths, and Workforces for operational agents, with options for on-premises and enterprise deployment.

These are architectural building blocks that can develop into multi-model systems. The actual advantage of each configuration over alternatives depends on the task, hardware, and operational testing.

Perhaps the most appealing human benefit is that users will less often need to know the AI's name.

Some tasks will become invisible within the phone or application: a phrase will be translated where translation is needed, a document understood where understanding is needed, and text corrected where an error occurs.

A sign of AI maturity may be that we stop feeling that we are using artificial intelligence every time we benefit from it.

Like electricity, which we no longer mention whenever we light a room.

When Arabic becomes an engineering advantage

Another aspect of small specialized models goes beyond memory and energy.

The difficulties posed by Arabic, with its morphological, dialectal, and semantic diversity, are not distributed evenly across tasks.

The problem faced by a system retrieving Arabic medical documents differs from the problem of recognizing recitation errors, from translating between Gulf and Maghrebi dialects, and from extracting tables from a scanned document.

A general model must represent a wide spectrum of these tasks within a single set of capabilities.

Specialization allows data, training, and evaluation to be directed toward the defined problem.

Take Kawn Embed Medical.

In medical documents, a patient may describe a symptom in colloquial language while the equivalent concept appears in the clinical record as a professional term. The job of a specialized embedding model is to bring these different expressions of the same meaning closer together, helping users find the relevant information.

It does not diagnose illness merely because it found suitable text; it performs a precise retrieval function that health applications need.

With Kawn Embed Islamic, the problem differs: connecting queries to relevant texts in the Qur'an, hadith, exegesis, and jurisprudence while preserving the importance of source attribution.

The two models may be similar in size, but their training data, semantic space, and success criteria differ.

I see in this flexibility broad scope for developing what might be called **hyperlocal specialized AI**.

These are models that understand dialects, professional expressions, and documents produced by particular communities without having to relearn all the world's knowledge.

They may develop more quickly than general models in their specialties because updating the data for a narrow task and evaluating its effect is sometimes more manageable than altering a comprehensive knowledge system.

This does not mean the giant has become useless.

Advanced general models will remain important for complex reasoning, unfamiliar situations, connecting knowledge domains, and serving as supervisors or fallback processors for difficult cases.

But in many designs they may become **the capability we turn to when necessary, rather than the capability we automatically run for the simplest things**.

That is a major change in design philosophy.

An economy that pays for genius only when it needs it

In its report on energy and artificial intelligence, the International Energy Agency estimated global data-centre electricity consumption at roughly 415 terawatt-hours in 2024; its later update projected that consumption would approach 950 terawatt-hours by 2030.

Artificial intelligence is not the sole cause of data-centre electricity consumption, but it has become one of the most important drivers of growth in demand.

As model use expands into every activity, a new question arises.

If millions of people use AI for small tasks throughout the day, is it computationally wise to send every task to the most capable and resource-intensive models?

The question resembles using a cargo aircraft to carry a paper letter.

The letter may arrive quickly, and the aircraft may be among engineering's finest products, but that does not make it the best means for every letter.

If a large share of daily tasks can be done locally or through smaller models, we can reduce pressure on computing infrastructure and reallocate heavy resources to work that truly merits them.

But this possibility does not automatically yield environmental savings. Cheap AI may lead to much greater use, and running multiple models without coordination may squander some benefits of shrinking them.

That is why I believe the next competitive field will not be merely reducing parameter counts, but improving **task-level economics**.

How many joules did the system consume before reaching a correct answer? How many milliseconds passed before the result appeared? What share of tasks required escalation to a larger model? What is the effect of error or delay on the user?

These measures open broad engineering possibilities.

We may find that the best system consists of a small model handling most requests, a more specialized one for unusual cases, and a large model that runs only when the problem becomes complex.

The router's job will resemble that of an orchestra conductor: it does not play every instrument, but knows when each voice should enter and when it should be silent.

A musical work does not improve by forcing every instrument to play at maximum volume.

It improves when each instrument has its place in time.

What do I expect by the end of this decade?

I believe the next few years will change our relationship with language models more deeply than current performance comparisons suggest.

The race for large models will continue, and progress toward more complex general capabilities will remain one of research's most important fields.

But alongside it, a quieter industry closer to daily life will grow: an industry of specialized models working in the background.

I see three connected shifts ahead.

First, **model selection will become part of intelligence itself**. An advanced system will not simply have the strongest model; it will know how to distribute tasks, predict their difficulty, estimate their cost, and choose the appropriate path.

Second, devices will gradually move from displaying the results of intelligence to carrying out a substantial part of it. The phone will not be merely a screen connected to a distant mind; it will have local capabilities that integrate with larger ones when needed.

Third, and most important in my view, will be the rise of what I call **cognitive fitness**.

Fitness here will not mean that a model knows more than everyone else. It will mean that a system can quickly adapt to a new task, add a specialty, replace one component with another, and improve one part without rebuilding the whole system.

A company with dozens of specialized models across sectors, dialects, documents, and vocal types may be better able to meet some market needs than a company offering only one general model, however powerful it is.

That is not because it has more models, but because the distance between the problem and the appropriate tool is shorter.

This is what I find worth watching in Misraj and Kawn over the long term.

If the team can turn its research family into an integrated system of small and medium models, with intelligent routing, continual evaluation, and local deployment capabilities, it will possess an engineering approach able to scale by adding specialties rather than merely enlarging one model.

Flexibility may become more important than size in many markets.

Institutions do not always need intelligence that can discuss everything; they need a system that understands their operations, acts within their limits, protects their data, and completes required work quickly, reliably, and at reasonable cost.

In that environment, owning a coordinated group of specialists may be a stronger advantage than owning one encyclopedic speaker.

At the scientific level, I believe the most important area for researchers will be the study of boundaries between these minds.

How can two models exchange knowledge without transmitting errors? How does a router know a task has left a model's specialty? How is a complete system tested while one of its components changes? Can small models develop an intermediate language for cooperation

that reduces the need to re-explain the entire context at every handoff?

These research questions are no less exciting than increasing parameter counts.

They may offer fertile ground for future Saudi and Arab discoveries.

A future more like a hive than a giant

I return to the bee swarm.

If you tried to explain its ability to choose a new home by studying the brain of a single bee alone, you would not reach the whole story.

You would need to understand the dance, the scouts' responses, the accumulation of information, the stop signals, and the threshold at which a possibility becomes a collective decision.

The swarm's intelligence is not merely the sum of bee brains.

It is also the way their relationships are organized.

This is how I see an important part of artificial intelligence's future.

The decisive question may not always be: what is the largest model we can run?

It may be: how do we build a system in which every component knows what it does well, what it does not do well, when it should work, and to whom it should hand the task next?

Raspberry Pi's small experiment in August 2026, research on model routing, and Misraj's family specialized in language, speech, documents, and retrieval are all distinct signs of a broader direction.

Artificial intelligence has not become small enough to solve everything on a limited device, nor have specialized models eliminated the need for giants.

But they have begun to change our conception of where intelligence should be placed and how much of it we need to complete useful work.

The most beautiful aspect of this direction may be that it opens the future to more varied forms of innovation.

Not every device needs to carry a supermind, not every model needs to know the whole world, and not every institution needs to run a capability it does not need.

Sometimes it is enough for a machine to understand its small task deeply, to find around it other machines that understand their tasks to the same degree, and for all of them to cooperate without forgetting their limits.

That is why I read Misraj's work from a different angle: not as an attempt to shrink the giant, but as an attempt to build **an ecology of specialized minds**, where capabilities can multiply and cooperate as living beings do in an ecosystem.

Perhaps a day will come when we do not know the name of the model that corrected a text, understood a document, helped a child read, or caught a fault in a machine.

We will only notice that things work more intelligently, that our privacy is no longer the necessary price of every form of assistance, and that small devices have learned to do what once seemed beyond them.

That day will not announce the death of giant models.

It will announce the maturity of artificial intelligence, when it stops trying to place the whole world inside one mind and learns how to distribute knowledge among many minds.

A bee does not need to become as large as the hive in order to serve it.

Great intelligence is not always knowing every answer; it is knowing which mind has the answer, how to reach it, and when wisdom lies in asking no one at all.

References and primary sources

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. [Kuwain 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.