

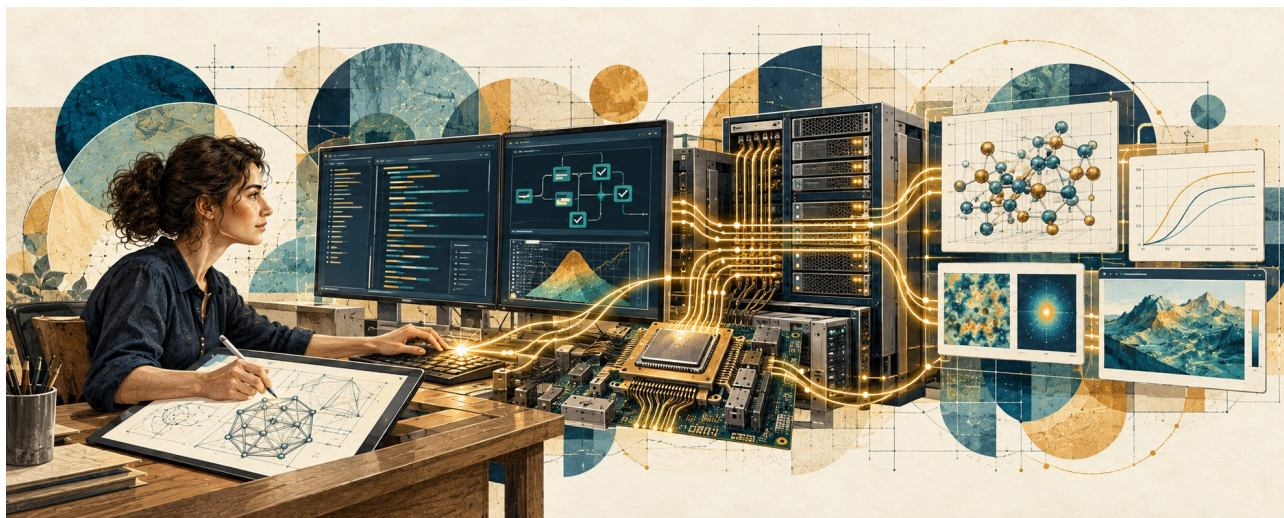
Making Sense of AI and the Acceleration Opportunity

Models, knowledge retrieval, and agents explained

LMM Labs

Lee Farretta and Matthew Farretta

Prepared in collaboration with ChatGPT



Opening illustration. Human-directed research, development, and publishing. AI-generated conceptual artwork.

Abstract

AI terminology has expanded faster than public understanding. Models, retrieval systems, agents, memory, and integration protocols do different jobs, yet they are often presented together. This review organizes the landscape around five functions: learning and generating, retrieving knowledge, taking action, coordinating work, and evaluating results. Together, these capabilities can accelerate research, software development, and publishing. The LMM Labs case study documents eleven applications deployed in Azure, including eight same-day initial working prototypes, alongside a reusable delivery platform. These results support substantial acceleration; the conventional-team comparison is an engineering estimate rather than measured labor savings. The practical opportunity is to turn more human ideas into working results and reuse what each project teaches.



Figure 1. The AI alphabet soup. An AI-generated conceptual illustration of overlapping terminology, not a technical hierarchy.

Introduction

Figure 1 captures a familiar experience: AI, ML, LLM, RAG, MCP, and other labels appear everywhere, often without an explanation of how they relate. What is a model? What makes an agent different? Where do retrieval, memory, and tools fit?

The confusion comes partly from mixing categories. A model generates an answer; a retrieval system finds information; an agent uses tools to carry out work. Other components organize the process or check the result. These capabilities can work together, but they are not interchangeable.

Our first task is to organize that vocabulary. Figure 2 groups the main capabilities by the jobs they do. Our second task is to explain the opportunity those capabilities create: accelerating the path from a human idea to useful research, software, or published work.

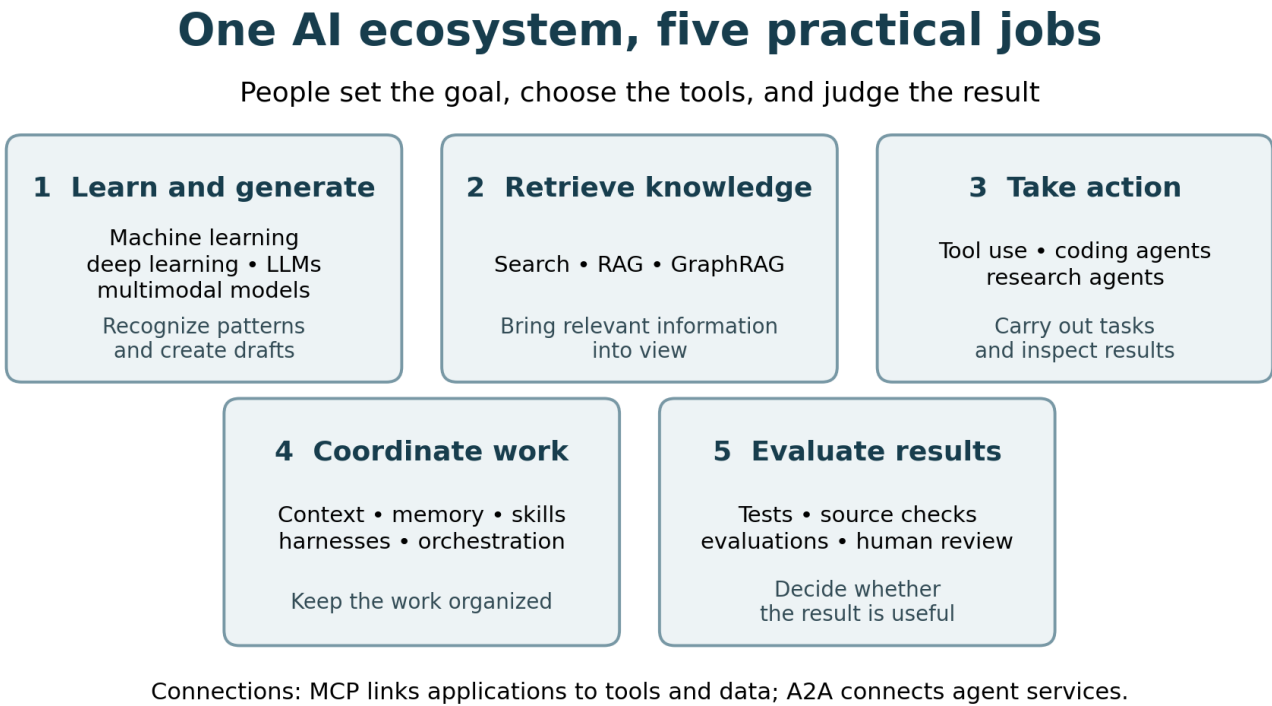


Figure 2. A practical map of the AI landscape. Categories overlap and are not a hierarchy or a required sequence. Original synthesis drawing on LeCun et al. (2015), Lewis et al. (2020), Yao et al. (2023), Anthropic (2024), and official MCP and A2A documentation.

1 From prediction to creation to action

AI is the broad field. Machine learning is one approach within it: systems learn patterns from data. Deep learning uses multilayer neural networks to learn more complex representations (LeCun et al., 2015). These methods still power forecasting, classification, and recognition.

Generative AI adds the ability to produce text, images, audio, and code. Large language models (LLMs) are a major part of this family. The Transformer architecture introduced by Vaswani et al. (2017) provided an important foundation for modern language models.

A model becomes more useful when it can draw on relevant information and work with tools. Retrieval and agents illustrate how those capabilities extend the model:

- **Retrieval.** External information can be brought into a model's task. Retrieval-augmented generation (RAG) combines retrieved material with generated answers (Lewis et al., 2020).
- **Agents.** Agents use tools and respond to what happens. They can search, edit files, run code, inspect results, and choose a next step. ReAct demonstrated a pattern that combines reasoning with action (Yao et al., 2023).

These approaches coexist. A forecasting model may need neither retrieval nor an agent. A document assistant may need retrieval but little autonomy. The right combination depends on the work.

That distinction takes us from the history of AI to the design of a working system. Instead of asking which label is newest, we can ask which capabilities a task needs and how they should work together.

2 Five jobs behind the alphabet soup

The five functions in Figure 2 explain the landscape without requiring a catalog of products:

- **Learn and generate.** Models recognize patterns, explain material, propose alternatives, and create drafts. Multimodal models work across formats such as text and images. A model can turn a rough idea into several alternatives, summarize a difficult document, or draft an explanation. This gives people material to assess and improve, often reducing the effort needed to begin.
- **Retrieve knowledge.** Search and RAG find relevant material. Reranking helps choose the best matches; graph-based retrieval can connect themes across a collection. For example, a research assistant can find passages in a set of papers before drafting an answer. The source remains available for checking. Retrieval is especially useful when the answer depends on specialized or changing information.
- **Take action.** Tools let a system do something beyond producing an answer. Coding and research agents combine these tools with decisions about what to try next. An agent might edit a program, run its tests, inspect a failure, and try a correction. The important change is that it can participate in the work itself, rather than leave every execution step to the user.
- **Coordinate work.** Context is the information available for the current task. Memory preserves information for later use. Skills package reusable instructions or procedures. Orchestration manages steps and handoffs. A project needs continuity as well as individual

answers. Retaining decisions and reusing good procedures can reduce repeated explanation, while coordination keeps dependent tasks in the right order.

- **Evaluate results.** Tests, source checks, monitoring, and human review establish whether the work meets its intended standard. Evaluation closes the loop. A draft must support its claims, and a program must behave as intended. Checking those outcomes tells the team which parts of the workflow are saving time and which need improvement.

A harness is the software around the model that manages tools, context, task state, and recovery. Saving a note as memory does not retrain the model. Adding a skill does not necessarily change its underlying capabilities.

Integration protocols connect these pieces. Model Context Protocol (MCP) helps applications access tools and resources. Agent2Agent (A2A) supports communication between agent services (A2A Project, n.d.; Model Context Protocol, n.d.). Think of them as connection standards, rather than new forms of intelligence.

These connections rely on ordinary engineering infrastructure: data ingestion, databases, deployment, and monitoring. The model is one part of the application. A small system may need only a few supporting components; Anthropic (2024) recommends starting with simple workflows and adding autonomy where it helps.

With that map in place, the frontier becomes easier to follow. The developments below extend the same functions into larger tasks: finding connections across more information, carrying out more steps, and testing more possibilities.

3 Where the frontier is moving

The frontier is moving from AI that supplies a response toward systems that can sustain useful work. Advances in knowledge access, software development, and scientific exploration show what that change makes possible:

- **Better knowledge access.** GraphRAG uses relationships and summaries to answer broad questions across document collections (Edge et al., 2025). Ordinary search or simpler RAG may still be the best choice for a narrow question. This matters when the question concerns a whole collection—such as recurring themes across hundreds of reports—rather than a single fact. The acceleration comes from reducing the effort needed to assemble a coherent view.
- **Software agents.** SWE-bench evaluates systems on real repository issues, moving beyond isolated code snippets (Jimenez et al., 2024). Agents can increasingly work through implementation and testing; a complete product still needs broader acceptance checks. A developer can define a bounded change and review the implementation and test results. That shifts some attention from producing every line of code to directing and evaluating the work.

- **Scientific exploration.** Co-Scientist generates and refines hypotheses under researcher direction, with experimental validation in selected biomedical applications (Gottweis et al., 2026). Its value comes from connecting proposed ideas to experiments. Researchers can examine more candidate explanations and decide which deserve laboratory work. Generating a hypothesis and validating it remain distinct steps; linking them is where the scientific value emerges.

Multimodal systems let these workflows use diagrams, screenshots, and other inputs alongside text. Persistent context maintains continuity, and multiple agents can divide work when specialization helps. Together, these capabilities make larger projects possible. The next question is how much acceleration they deliver in practice.

4 What the evidence says about acceleration

Published studies already demonstrate substantial acceleration in writing, customer support, and consulting. Table 1 summarizes those findings alongside a software-maintenance study with a different outcome. Each measures a particular task and tool generation; none defines the full potential of today’s agentic development workflows.

Table 1. Selected productivity evidence

Work and source	Reported result	What the result covers
Professional writing Noy & Zhang (2023)	40% less task time; 18% higher rated quality	453 professionals; selected writing tasks
Customer support Brynjolfsson et al. (2025)	15% more issues resolved per hour, on average	5,172 agents; gains varied across workers
Consulting Dell’Acqua et al. (2026)	25.1% less time and 12.2% more tasks within the tested capability frontier	758 consultants; performance worsened on a task outside that frontier
Software maintenance Becker et al. (2025)	19% longer task time with early-2025 AI tools	16 experienced developers; 246 tasks in familiar repositories

Note. Results use different measures and task settings. Publication dates can differ from the dates of the experiments. These are study findings, not estimates of LMM Labs performance.

The software-maintenance result concerns experienced developers working in familiar repositories with early-2025 tools. A 2026 follow-up reported indications of speedup but could not establish a reliable estimate (Becker et al., 2026). That finding identifies the limits of a particular experiment. It does not negate the acceleration documented in other work or establish a ceiling for newer systems.

The larger opportunity comes from connecting capabilities across a complete workflow. When AI helps with requirements, implementation, testing, infrastructure, and release, it can shorten

the path from an idea to a running application. LMM Labs provides a concrete example of that broader change.

5 LMM Labs case study of accelerated development

LMM Labs delivered eleven application prototypes to Azure using human-guided AI and agentic automation. Eight initial working prototypes were completed the same day. The effort also created a reusable architecture and cloud delivery platform. The case study supports a clear finding: substantial acceleration extended across application construction and the engineering needed to deliver it (LMM Labs, 2026). The review covers initial construction and releases from August 1 through September 8, 2026.

Table 2. LMM Labs development and delivery results

Area	Documented result	Interpretation
Applications	Eleven of eleven deployed in Azure.	Working prototype portfolio; live inventory verified September 29.
Initial construction	Eight same-day prototypes; Palate about 23 hours; Stock 1–2 days with notebook reuse; Songsmith about two days.	Short build cycles across varied domains. First prototypes can precede further refinement.
Shared platform	Initial cloud pipeline August 7–8; container migration August 24–26.	Architecture, scripts, and release patterns reused across applications.
Tax engineering	First reported working prototype in about 14 minutes; expanded scope developed across five dates, August 13–18.	Database, tenant isolation, accounts, document workflows, and tests added before release.
Release execution	Selected application jobs: 4 min 28 sec to 9 min 41 sec. Full sessions reportedly took a few hours, sometimes 4–5.	Recorded job runtimes and owner-reported sessions measure different parts of delivery.

Tax shows the difference between a fast first prototype and deeper engineering. Its later rounds added multi-user data, identity, and document capabilities. The Tax-only conventional estimate is 30–55 person-days, included within the suite total. The shared platform is another part of the accelerator: agents helped build the infrastructure that made subsequent releases repeatable.

The conventional comparison estimates 143–270 person-days, or 1,144–2,160 hours, for equivalent prototypes and shared delivery capability. That represents approximately 7–14 working weeks for four full-time contributors under ideal allocation. The estimate includes implementation, requirements, architecture, persistence and integrations, infrastructure, testing, and documentation. Shared platform work is counted once; the comparison retains modern frameworks, managed cloud services, and CI/CD.

The evidence establishes rapid construction and delivery. Actual working hours and a matched conventional build were not measured, so the planning estimate does not establish an exact labor saving or speedup multiplier. Intentional release holds are separate from build time.

The practical gain is also one of access. Lee Farretta reports that the portfolio would previously have required months of work and might not have been feasible for him to complete. AI made it possible to turn his ideas and technical direction into running applications on short cycles. Reusing the platform then reduced the setup required for each subsequent project.

6 Keep people at the center

Human direction made this portfolio possible. People supplied the ideas, domain knowledge, priorities, and acceptance decisions; agents performed much of the implementation and supporting engineering. As those capabilities expand, a small team can undertake work that previously required several specialized contributors. That changes what people can build, how teams organize work, and where human judgment matters most.

Useful safeguards can remain practical: verify sources in an article, test intended behavior in software, and give agents permissions suited to their task. NIST identifies information integrity, privacy, and security among the risks that need attention (Autio et al., 2024). These checks help preserve the gains by reducing avoidable rework.

Conclusion

The AI landscape is easier to understand when its labels are connected to useful work. Models generate possibilities, retrieval supplies information, agents take action, coordination maintains continuity, and evaluation checks the result. These are complementary capabilities that people can assemble around a clear purpose.

The LMM Labs case study makes the opportunity concrete: eleven applications deployed, repeated same-day initial prototypes, and a reusable delivery platform. Published studies provide additional evidence of gains in other settings. Substantial acceleration is already visible in the work delivered.

The lasting opportunity goes beyond finishing one task faster. Faster execution makes room for more experiments and brings projects within reach that might otherwise remain ideas. When

reusable architecture, tools, and lessons improve the next project, acceleration compounds across research, development, and publishing.

Research and preparation notes

This narrative review draws on foundational research, empirical productivity studies, primary technical documentation, and the LMM Labs descriptive case study, with a research cutoff of September 29, 2026. The case study reconstructs build and release milestones from retained sessions, source records, Git history, GitHub Actions, Azure metadata, and validation reports. It is not a controlled experiment. Preprints and vendor guidance are identified in the references. Figure 2 is an original synthesis of the concepts discussed in the article.

ChatGPT assisted with research synthesis, drafting, editing, and visual preparation. LMM Labs supplied editorial direction and application-development evidence. Codex produced the conventional-effort planning estimate. © 2026 LMM Labs.

References

- A2A Project. (n.d.). *Agent2Agent protocol specification*. Retrieved September 29, 2026, from <https://a2a-protocol.org/latest/specification/>
- Anthropic. (2024, December 19). *Building effective agents*. <https://www.anthropic.com/engineering/building-effective-agents>
- Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). *Artificial intelligence risk management framework: Generative artificial intelligence profile* (NIST AI 600-1). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.AI.600-1>
- Becker, J., Rush, N., Barnes, E., & Rein, D. (2025). *Measuring the impact of early-2025 AI on experienced open-source developer productivity* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2507.09089>
- Becker, J., Rush, N., Cunningham, T., Rein, D., & Mahamud, K. (2026, February 24). *We are changing our developer productivity experiment design*. METR. <https://metr.org/blog/2026-02-24-uplift-update/>
- Brynjolfsson, E., Li, D., & Raymond, L. R. (2025). Generative AI at work. *The Quarterly Journal of Economics*, 140(2), 889–942. <https://doi.org/10.1093/qje/qjae044>
- Dell’Acqua, F., McFowland, E., III, Mollick, E., Lifshitz, H., Kellogg, K. C., Rajendran, S., Kraye, L., Candelon, F., & Lakhani, K. R. (2026). Navigating the jagged technological frontier: Field experimental evidence of the effects of artificial intelligence on knowledge worker productivity and quality. *Organization Science*, 37(2), 403–423. <https://doi.org/10.1287/orsc.2025.21838>
- Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2025). *From local to global: A graph RAG approach to query-focused summarization* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2404.16130>

- Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Sirkovic, P., Myaskovsky, A., Glowaty, G., Weissenberger, F., Orlandi, A., Popovici, D., Palepu, A., Rong, K., Tanno, R., Saab, K., Zhang, F., Blum, J., Carroll, A., Kulkarni, K., Tomašev, N., . . . Natarajan, V. (2026). Accelerating scientific discovery with Co-Scientist. *Nature*, 655, 487–496. <https://doi.org/10.1038/s41586-026-10644-y>
- Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., & Narasimhan, K. (2024). *SWE-bench: Can language models resolve real-world GitHub issues?* International Conference on Learning Representations. <https://openreview.net/forum?id=VTF8yNQM66>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459–9474. <https://papers.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- LMM Labs. (2026). *Evaluating the initial LMM Labs methodology: A case study of rapid application development with human-guided AI and agentic automation* [Internal case study]. Evidence verified September 29, 2026.
- Model Context Protocol. (n.d.). *Architecture overview*. Retrieved September 29, 2026, from <https://modelcontextprotocol.io/docs/2026-07-28/learn/architecture>
- Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654), 187–192. <https://doi.org/10.1126/science.adh2586>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. <https://papers.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). *ReAct: Synergizing reasoning and acting in language models*. International Conference on Learning Representations. https://openreview.net/forum?id=WE_vluYUL-X

Glossary

- **A2A — Agent2Agent.** A protocol that lets separate agent services communicate and coordinate tasks.
- **Agent.** An AI system that uses tools, observes results, and chooses its next step toward a goal.
- **AI — Artificial intelligence.** The broad field of building systems that perform tasks involving learning, reasoning, perception, or action.
- **API — Application programming interface.** A defined way for software systems to exchange requests, data, and results.
- **Azure.** Microsoft's cloud platform, used to host the LMM Labs applications.
- **CI/CD — Continuous integration and continuous delivery or deployment.** Automation that builds, checks, and prepares or deploys software changes.
- **Context.** The instructions, conversation, documents, and other information available to a model for its current task.
- **DL — Deep learning.** Machine learning that uses neural networks with multiple layers.
- **Generative AI.** AI that produces content such as text, images, audio, or code.
- **Git and GitHub.** Git records changes to source files; GitHub hosts repositories and supports collaboration and automated workflows.
- **GPT — Generative pre-trained transformer.** A family of generative models built on the Transformer architecture and pre-trained on large datasets.
- **GraphRAG.** Retrieval that uses a knowledge graph and summaries to connect information across a document collection.
- **Harness.** The software around a model that manages tools, context, task state, and recovery.
- **Knowledge graph.** A representation of entities and the relationships between them.
- **LLM — Large language model.** A model trained on large amounts of text to process and generate language.
- **LMM Labs.** The research and development team behind this article and its application case study.
- **MCP — Model Context Protocol.** A standard for connecting AI applications to tools and information sources.
- **Memory.** Information preserved so a system can use it in later tasks or interactions.
- **ML — Machine learning.** Methods that learn patterns from data rather than relying only on explicitly programmed rules.
- **Multimodal model.** A model that works with more than one type of input or output, such as text and images.
- **Neural network.** A model made of connected computational units whose adjustable weights are learned from data.

- **NLP — Natural language processing.** Methods for analyzing, understanding, or generating human language.
- **Orchestration.** Managing the steps, dependencies, and handoffs in a workflow involving models, tools, or agents.
- **Persistence.** Storing application data so it remains available after a session ends or a service restarts.
- **Person-day.** Eight working hours of one person's effort in the case-study estimate; distinct from elapsed calendar time.
- **Prototype.** An initial working application used to explore an idea and test its capabilities.
- **R&D — Research and development.** Work that investigates ideas and turns them into useful methods, experiments, or products.
- **RAG — Retrieval-augmented generation.** Combining retrieved source material with a model's generated response.
- **Reranking.** Reordering retrieved results to prioritize those most relevant to a query.
- **RL — Reinforcement learning.** Learning through feedback or rewards associated with actions and their outcomes.
- **Skill.** A reusable set of instructions or procedures that guides a system through a task.
- **Transformer.** A neural-network architecture that uses attention to model relationships within its input.