ChatGPT

Agentic AI and Agentic Systems

What Is Agentic AI? (High-Level Overview)

Agentic AI refers to artificial intelligence systems that act as autonomous *agents* – they can make independent decisions and take actions to achieve goals with minimal human supervision [1] [2]. In simple terms, an agentic AI is goal-driven, proactive, and adaptive, rather than just following fixed rules or responding passively. The term "agentic" highlights that these AI systems possess **agency** – the capacity to **act independently and purposefully** toward objectives [1] [2]. This contrasts with traditional software or even basic machine learning models that only operate within predefined constraints or require explicit step-by-step instructions from humans [3].

Unlike a static **generative AI** model (e.g. a text generator that only outputs content when prompted), an agentic AI system *builds upon* generative capabilities by **actively using its outputs to pursue specific goals in a dynamic environment** [4]. For example, a generative model like ChatGPT can produce an answer about "the best time to climb Mt. Everest," but an agentic AI could **take it a step further** – not only informing you of the best time, but also **planning the trip and booking the flight and hotel autonomously by calling external tools** [5]. In essence, agentic AI turns AI outputs into goal-oriented actions.

Agentic systems typically consist of one or more AI agents (hence "agentic system") that may work together. In fact, the terms **"agent"** and **"agentic system"** are often used interchangeably in literature. *"Agent"* usually refers to the autonomous software entity itself, whereas *"agentic system"* might emphasize the system as a whole (including the agent along with its tools, environment interfaces, etc.) [6]. Both imply an AI that exhibits autonomy, goal-directed behavior, and the ability to perceive and act within an environment to accomplish tasks.

Formal Definitions from AI Literature

In classical AI literature, an **agent** is often defined in terms of how it **perceives and acts**. A widely cited formal definition (from *Russell & Norvig's* framework) is: *"An agent is anything that can perceive its environment through sensors and act upon that environment through actuators (effectors)."* [7] In other words, an agent has some form of **perception** (to take in information about the world or context) and some form of **action** mechanism to affect the world. This definition spans a broad range: a robot has physical sensors and motors as actuators, whereas a software agent might "sense" via API calls or data inputs and act by outputting answers or triggering software commands [7].

Additionally, agents are often discussed in terms of **rationality** and goal-driven behavior. A *rational agent* is one that acts to **maximize its expected performance measure** given the information it has and the goals it needs to achieve [8]. The agent's behavior can be described by an **agent function** mapping percepts (inputs) to actions that progress it toward its goals [9]. Key properties that characterize an intelligent agent include: autonomy (operating without constant intervention), **goal-oriented** behavior (seeking to achieve objectives or maintain desired states), **reactivity** (responding to changes in the environment), and often **proactiveness** (taking initiative to fulfill goals). Modern definitions of *agentic AI* echo these concepts, emphasizing **autonomy, decision-making, and the ability to perform multi-step tasks towards a goal** [1] [3].

It's worth noting that contemporary usage (e.g. in industry) aligns closely with these formal ideas. For instance, IBM defines agentic AI as an AI system that **"exhibits autonomy, goal-driven behavior and adaptability,"** in which multiple AI agents might handle different subtasks and coordinate towards the overall goal [10] [2]. Similarly, Amazon describes agentic AI as an **autonomous AI that can act independently to achieve pre-defined goals**, highlighting that *"'agentic' indicates agency – the ability to act independently in a goal-driven manner."* [1]. These definitions reinforce the classical view of agents but in the context of current AI (often involving learning models and complex environments).

Core Components of Agentic Systems

While implementations vary, most agentic AI systems share a set of **core components or capabilities** that enable their functioning. These map onto the classical notion of an agent (perceive, decide/plan, act) with some additional features like memory. Below is a breakdown of these core components:

Perception (Sensing the Environment)

Perception is how an AI agent gathers information about the state of the world or context in which it operates. An agent must have up-to-date data to make good decisions. In practical terms, this could mean reading sensor inputs, receiving user queries, or fetching data from APIs and databases [11]. For example, an agent might take in text input from a user, or a financial agent might continuously ingest market data streams. Agentic AI "begins by collecting data from its environment through sensors, APIs, databases or user interactions," ensuring it has current information to analyze [11]. Advanced agents can handle various data types – a web-based agent might **call APIs, perform web searches, or do OCR on documents** to extract relevant facts [12] [13]. This perceptual step mirrors the human senses: it's how the agent *sees/hears/reads* the world.

Reasoning (Interpreting and Deciding)

Once information is perceived, an agent engages in **reasoning** – interpreting the inputs, drawing inferences, and determining what it means for the task at hand. Using AI techniques (natural language understanding, pattern recognition, etc.), the agent "processes [the data] to extract meaningful insights," understands the context or user intent, and detects important patterns [14]. This reasoning ability allows the agent to figure out *what* needs to be done or to evaluate the situation. Modern agentic systems often leverage powerful models (like language models or other AI components) at this stage to comprehend instructions and contextual information. For instance, an agent might use an LLM to parse a complex query and figure out the necessary steps to fulfill it [15]. Reasoning also involves handling ambiguities or errors gracefully and updating interpretations as new data comes in [16]. In essence, this component is the agent's "brainwork" – making sense of inputs and deciding on a course of action based on its goals.

Planning (Goal Setting and Strategy)

In agentic AI, **planning** is the step where the agent formulates its objectives and devises a strategy or sequence of actions to achieve them. After reasoning about the situation, the agent will set sub-goals or choose a high-level plan that leads toward fulfilling the overall goal [17]. This often involves search or decision algorithms to figure out the best path. For example, the agent might break a complex task into smaller subtasks or choose between different possible methods to proceed. Formally, this relates to the field of AI planning and decision-making: the agent might construct a plan using decision trees, state-space search, or even learned policies (reinforcement learning) to reach the goal state [17]. In simpler terms, planning is the agent asking *"What steps should I take to get this done?"* and organizing those steps. A sophisticated agent can **dynamically re-plan** if conditions change or if its initial approach fails, showing adaptability. This component ensures the agent's actions are not random but **purposefully directed toward goals**.

Action (Executing and Effecting Change)

After planning, the agent takes **action**. The **action component** (sometimes called *execution*) is how the agent's decisions actually interface with the outside world. An agent can act in various ways depending on its embodiment: a software agent might call external services, execute code, query a database, or output text to a user; a physical robot agent might move motors or speak. For agentic AI, actions often involve **tool use** – e.g. calling an API, running a plugin, or controlling an application to carry out a task [18] [19]. Crucially, agent actions can be multi-step and iterative. The agent might execute a series of actions (with loops of perceiving results and adjusting as needed) to complete a complex task. For instance, an agent tasked with booking travel will perform actions like searching for flights, comparing options, then making a reservation via a web API. Modern AI agents can even write and run code, or interact with documents/software, as part of their action repertoire [20]. This ability to *affect* the environment is what distinguishes an agentic system from a passive AI. All actions are typically monitored or logged (for safety and oversight), and some agent frameworks include a **human-in-the-loop** for critical actions (requiring human approval before execution) [21] [22].

Memory (State, Learning, and Adaptation)

Memory is a key component that allows an agent to maintain state and improve over time. There are a few facets of memory in agentic systems:

- **Working Memory / Context:** This is the short-term memory that an agent uses during a task, such as the history of dialogue in a conversation or the sequence of recent percepts. For example, language-model-based agents keep track of the conversation or instructions given so far. This context memory ensures the agent's actions remain coherent and consistent. In fact, many agent frameworks equip LLMs with **long-term memory buffers or databases** so that the agent can recall facts or decisions made earlier and not repeat mistakes [23]. AWS notes that at the reasoning stage, *"LLMs use long-term memory systems to ensure that situational and context-dependent tasks remain consistent throughout the entire process."* [23].

- **Long-Term Knowledge / Learning:** An agent can also have a knowledge base or learned model that it updates. Agentic AI often involves **learning from experience**: after each action or episode, the agent can receive feedback and adjust its behavior. This corresponds to techniques like reinforcement learning or self-supervised updates. For instance, an agent might learn from successes and failures by adjusting its strategy to perform better next time [24]. Some agents continuously refine themselves by storing new information (outcomes, user preferences, etc.) in a knowledge store. In multi-agent setups, agents might even share a **communal memory** to propagate learnings across the system [25].

- **World State Tracking:** For complex environments, an agent maintains an internal state or model of the world (which can be considered a form of memory) to know what has changed. This is analogous to how a game-playing AI remembers the current game state. It helps in reasoning about what to do next.

In summary, memory allows an agent to be **context-aware** and **improve over time**, rather than treating every situation as a blank slate. An agent with memory can, for example, remember a user's preferences from earlier interactions or learn from mistakes (avoiding them in the future). With the right feedback loops, agentic systems can thus become more effective and personalized through continuous learning [24].

LLMs and SLMs as AI Agents

Modern agentic AI has been greatly enabled by advances in AI models – especially **Large Language Models (LLMs)**. Today, many AI agents use LLMs as the "brain" for reasoning and decision-making [26]. An LLM (like GPT-4, etc.) can take on the reasoning/planning role by interpreting instructions, breaking down problems, and deciding which actions to take. In fact, *"the core components powering most modern AI agents are (very) large language models"*, as one recent survey notes [26]. These models provide the **intelligence** that lets agents choose tools, control task flows, decompose complex tasks into subtasks, and perform the reasoning for planning and problem-solving [26]. Essentially, an LLM can serve as the agent's **mind**, figuring out what needs to be done and even generating code or API calls as actions.

However, it's important to clarify: an **LLM by itself is not necessarily an "agent."** A plain LLM (such as ChatGPT in its base form) is generally **reactive, not autonomous** – it generates responses when prompted by a user, but it doesn't set its own goals or take actions without instruction [27] [28]. As one explainer puts it, *"ChatGPT is not a true AI agent… It responds to your prompts but doesn't take action on its own"* [27] [28]. It lacks the *self-directed goal pursuit* and *initiative* that define agency. That said, the line can blur: LLM-based systems can appear agent-like (for instance, they can carry out a complex user request by generating a multi-step solution). But to transform an LLM into a **proper agent**, it must be embedded in a loop that lets it plan steps and act (often via tools) autonomously. This is exactly what frameworks like **LangChain, AutoGPT, and similar "LLM agent" systems** do – they give the LLM a mechanism to iteratively decide, act (e.g. by calling a tool or triggering code), observe the result, and then decide again, until a goal is reached [29]. In these setups, the LLM is a component (the reasoning module) of a larger agentic system that provides memory, tool interfaces, and sometimes an orchestrator to manage the process.

Do Large Language Models qualify as agents? In summary: *not on their own,* but **yes when they are part of an agentic loop**. If the workflow gives an LLM the freedom to set intermediate goals, choose actions, and carry them out (with the ability to utilize tools and observe outcomes), then the LLM effectively operates as an agent [30]. For example, GPT-4 with the proper prompting and plugins can search the web, use calculators, write and run code, etc., making decisions along the way – this begins to **cross into agentic behavior** rather than just Q&A chatting [31]. Many "autonomous AI" demos are essentially an LLM given a high-level goal and then looping its own outputs to achieve subgoals (e.g. the *AutoGPT* platform sets a goal for GPT-4 and it spawns a chain of actions to fulfill it) [29]. So, LLMs *can* be central to agents, but they need that surrounding infrastructure to manifest true agency.

The Rise of Small Language Models (SLMs) in Agents

While early agent systems often used very large models, there is a growing recognition that **smaller models can often do the job**. A notable recent work titled *"Small Language Models are the Future of Agentic AI"* (Belcak *et al.*, 2025) argues that **Small Language Models (SLMs)** – which the authors define roughly as models **small enough to run on common consumer devices with low latency** (typically under ~10 billion parameters as of 2025) – are *"sufficiently powerful, inherently more suitable, and necessarily more economical"* for many agentic tasks [32] [33]. In their view, the current industry habit of using one giant LLM for every agent task is inefficient. Many AI agents perform *"a small number of specialized tasks repetitively and with little variation,"* so a lightweight model fine-tuned for those specific tasks can **handle them just as well** [32] [34]. Because these tasks are often narrow (e.g. formatting data, making an API call with specific parameters, generating code in a particular style), an SLM that is trained or fine-tuned for the narrow domain can be both **more efficient and more reliable** (less likely to go off-script) than a huge general model [34]. Moreover, using SLMs brings big practical benefits: lower latency (faster responses), lower computational cost, and the ability to deploy on-premises or on-device (enhancing privacy and reducing dependency on costly cloud inference) [34]. In short, **for many agentic systems, smaller specialized models "not only suffice, but are often preferable."** [34]

The push for SLMs is also about **economy and scalability**. Running a 70B+ parameter model for every agent action is resource-intensive and expensive, which doesn't scale well when an agent might make thousands of calls. By contrast, a network of many small models each handling a portion of tasks could be far more cost-effective. The authors of *"Small Language Models are the Future of Agentic AI"* even outline methods to systematically replace LLMs with SLMs in existing agent frameworks [35] [36]. They also note an ancillary benefit: it could democratize agent development, since smaller models are easier for individuals and smaller organizations to train and deploy (broadening participation and innovation in the field) [37].

That said, SLM advocates acknowledge that **large models still have an edge in general-purpose reasoning and open-ended conversation**. For cases where a broad, human-like intelligence or dialog is needed, a large model might outperform a small one [38] [39]. The solution? **Heterogeneous agentic systems.** Instead of one monolithic LLM, an agentic system can use *multiple models of different sizes* for different subtasks [40] [41]. For example, a complex agent might use a small model by default for routine tasks and only call a big LLM when a very general conversation or complex reasoning is required [42]. Belcak *et al.* argue that such *"SLM-first, LLM-sometimes"* architectures combine the precision and efficiency of SLMs with the broad capabilities of LLMs, yielding agents that are both **cost-effective and capable** [42]. In fact, an agent could even call an LLM as just another tool when needed – conceptually, *"a language model can itself be a tool called by another language model,"* allowing a hierarchy where the heavy model is only invoked sparingly [43].

In summary, LLMs have proven extremely powerful for enabling agentic AI, but they are not the only solution. **Small Language Models** are increasingly viable as the core of agents, especially for well-defined or repetitive tasks, providing a more sustainable path for deploying large numbers of agents [32] [34]. With clever orchestration (possibly mixing model sizes), we can build agentic systems that leverage big models' strengths only when necessary and otherwise rely on efficient specialized brains. As the NVIDIA researchers put it, with today's techniques *"capability – not parameter count – is the binding constraint,"* meaning **a well-**

designed 5B or 10B model can often match the performance of a much larger model on the narrow tasks an agent needs to do [44] . This trend foreshadows a future ecosystem of many **agentic AI services powered by a federation of smaller models** rather than a few giant monoliths.

Examples of Agentic AI in Education, Healthcare, and Research

To make these concepts more concrete, here are a few examples of how LLM-based or SLM-based agents can be used in different domains:

- **Education:** Agentic AI is being explored to build intelligent tutoring systems and teaching assistants. For instance, an *LLM-powered educational agent* could monitor a student's progress, personalize the curriculum, and provide feedback or hints in real time. Such an agent perceives the student's inputs (answers, questions), **retains memory** of the student's learning history, and reasons about what concept to teach or review next. It might plan a sequence of exercises tailored to the student's needs and act by generating explanations or quizzes. Recent research surveys note that LLM-based agents in education leverage memory to **store long-term knowledge about a student's habits and context**, use tools to retrieve educational content or perform tasks like grading, and plan optimal learning paths or strategies for the student [45] [46] . For example, an agent might automate feedback on homework by first generating initial comments on a student's essay (Agent 1) and then having another agent review and refine those comments (Agent 2) to ensure accuracy and clarity [47] . In short, agentic AI tutors can adapt to each learner, providing interactive and personalized instruction beyond what static e-learning systems offer.

- **Healthcare:** In medicine and healthcare, agentic AI can serve as an assistant to clinicians or patients. Imagine a healthcare agent that continuously monitors patient data (vitals, lab results, symptoms) and helps manage treatment plans. Such an agent could **perceive** updates in a patient's health records or real-time sensor data, **reason** about potential issues or needed adjustments (e.g. noticing a lab value out of range), and **act** by alerting medical staff or even adjusting a treatment parameter. For example, an agent might detect that a patient's blood pressure is rising and proactively adjust medication dosage or schedule an earlier check-in. IBM gives a scenario: *"In healthcare, agents can monitor patient data, adjust treatment recommendations based on new test results, and provide real-time feedback to clinicians through chatbots."* [48] . This means an agent could talk to doctors via a chatbot interface, explaining its findings or suggestions, or even interface with hospital systems to schedule tests or flag urgent conditions. Another use is patient-facing agents – like an AI health coach that answers questions and gives guidance using up-to-date medical knowledge (retrieving information from medical databases when needed). Because healthcare is high-stakes, these agents are usually designed to work *with* human oversight, but they can greatly streamline tasks (e.g., triaging patient inquiries, summarizing patient history for a doctor, ensuring follow-up steps are taken).

- **Scientific Research:** Research involves sifting through vast literature, planning experiments, and aggregating results – tasks well-suited for agentic AI. A research assistant agent might **read** new papers or database entries (perception), **synthesize** findings to derive insights (reasoning), and then **plan** and propose next steps, such as suggesting an experiment or drafting a summary report (action). Because it can access external knowledge sources, such an agent can stay up-to-date. For instance, an agent could automatically search for the latest publications on a topic and summarize the state of the art for a scientist. AWS describes that an agentic AI in R&D could *"draw from recent research published on credible platforms, synthesize the results, plan further tests, and present* *researchers with the final product they need to investigate."* [49] . This might save huge amounts of time in literature review and experiment planning. In practice, there are early examples: an agent that helps chemists by planning synthesis routes for new compounds (using databases of chemical reactions), or an agent that assists in writing research drafts by collating relevant references and ensuring consistency. In essence, for knowledge-intensive and repetitive parts of research, agentic AI can act as a tireless assistant – gathering data, connecting dots, and even running simulations – allowing human researchers to focus on creative and critical evaluation.

Agentic AI in the Broader AI Ecosystem

Agentic AI sits at the intersection of many subfields of AI and inherits concepts from each. To understand its place, it's useful to see how it relates to **multi-agent systems, task-oriented agents, and retrieval-augmented systems**, among others:

- **Multi-Agent Systems:** Agentic AI need not be about just one agent working in isolation. Often, we deploy **multiple agents that collaborate or coordinate**, which is the focus of multi-agent systems (a long-standing field in AI). In a multi-agent setup, different agents might handle different responsibilities or work in parallel on subproblems. This can **mirror a team structure**: for example in a business process, one agent could specialize in data collection, another in analysis, and a third (a higher-level agent) in making final decisions. Multi-agent systems can be organized in different structures. A *horizontal* (decentralized) structure means agents are peers, each specialized in a certain skill, and they communicate laterally to solve the problem together [50] . A *vertical* (hierarchical) structure means there's a chain of command – e.g. a top-level agent (or "conductor") orchestrates the high-level reasoning and delegates simpler tasks to lower-level agents [51] [52] . Each lower agent might do a micro-task (like calling an API or formatting data) and pass results up the chain. Multi-agent approaches are powerful for complex or scalable problems: *"multi-agentic AI… involves multiple AI agents collaborating to break down complex workflows into smaller segments,"* which is more flexible and scalable for complex scenarios [53] [54] . Real-world examples include **MetaGPT**, a framework where different agents take roles like "PM," "Developer," "Tester" to collaboratively generate software; or a customer service system where one agent analyzes sentiment while another fetches account data and another composes a response. The agentic AI concept encompasses these multi-agent orchestration scenarios as a natural extension – in fact, coordinating many agents introduces challenges of its own (communication, conflict resolution) that researchers address in multi-agent theory.

- **Task-Oriented Agents:** A large subset of agentic AI applications are essentially **task-oriented agents** – agents designed to accomplish specific tasks or services. This term is often used in contrast to open-ended conversational agents. For instance, a *task-oriented dialogue agent* is one that helps a user achieve a goal (book a flight, fix a software issue, schedule a meeting) through conversation, as opposed to just chit-chat. In the broader sense, any agent that is **purpose-built for a particular task or domain** (and sticks to that) can be considered task-oriented. These agents leverage their autonomy and planning to carry out well-defined procedures: e.g., an agent that autonomously handles IT support tickets (diagnosing the issue and providing a fix) or an agent that plans a personalized workout routine and nutrition plan for a user. **Agentic AI is often inherently task-oriented**, since by definition it is pursuing goals. We see this in how agentic AI is deployed in automation use cases – the agent is given a clear objective (solve this problem or complete this workflow) and it utilizes its perception, reasoning, and action capabilities to get it done. By focusing on tasks, such agents can also be optimized or fine-tuned specifically for those tasks (this is where SLMs shine, as they can be domain-experts). Notably, frameworks like LangChain facilitate building task-oriented agents by allowing developers to define what the agent needs to achieve and equipping it with tools accordingly. In summary, within the AI ecosystem, task-oriented agents are the pragmatic side of agentic AI – they are **the doers**, taking the impressive raw abilities of AI models and channeling them into executing real-world tasks (from answering customer queries to controlling IoT devices).

- **Retrieval-Augmented Agents:** One challenge with using learned models (like LLMs) is that their knowledge can be stale or limited to their training data. **Retrieval-Augmented Generation (RAG)** is a popular technique where the model is supplemented with an external knowledge source – essentially, *it retrieves relevant information on the fly* and uses it to produce more accurate outputs. In the context of agents, we talk about **retrieval-augmented agents**: these are agentic systems that integrate search engines, databases, or knowledge bases into their perception and reasoning steps. Instead of relying solely on the model's internal memory, the agent can **pull in fresh or specific data** as needed [55]. For example, an agent answering financial questions might query the latest stock prices or company reports via an API (ensuring it's using up-to-date info, not just what it saw in training). Another might use a vector database to fetch relevant documents (say, a specific policy file) to help it complete a task. By augmenting retrieval, the agent becomes much more powerful: it combines the **critical thinking of AI** with the **factual grounding of a database**. A practical illustration is a chatbot agent that, when asked a question, will first search the company's internal wiki for the answer and then craft a response – the agent autonomously decides to use the search tool as an action during its reasoning stage. IBM notes that *LLMs alone can't directly interact with external data or tools, but agents can — they "can search the web, call APIs and query databases, then use this information to make decisions and take actions."* [19]. This ability to incorporate live information not only improves accuracy but also allows agents to operate in changing environments (where new data continuously comes in). In the broader AI landscape, retrieval-augmented agents blur the line between static learned intelligence and dynamic knowledge retrieval, giving us systems that are both **knowledgeable and up-to-date**.

Finally, it should be mentioned that **agentic AI** overlaps with other concepts like **embodied AI** (agents with a physical presence, such as robots or autonomous vehicles, which perceive via sensors like cameras and act via motors). An autonomous car, for example, is an agentic system in the physical world: it perceives the road through cameras/LiDAR, reasons about driving, plans a route, and acts by steering and braking – a quintessential agent with all the components we discussed. In software realms, we also have **agent-based modeling** (where many simple agents interact to produce complex phenomena) and **reinforcement learning agents** (which learn optimal actions through trial-and-error). The current surge of interest in language-based agents is bringing these strands together, enabling *tool-using*, *collaborative*, *goal-driven* AI systems that operate in digital environments and beyond.

In summary, Agentic AI represents a shift from AI that is merely *descriptive or generative* to AI that is **active and goal-seeking**. Whether it's a single AI assistant helping with your email (task-oriented), a team of AIs running a business process (multi-agent), or a self-improving research aide that reads the internet (retrieval-augmented), agentic systems are poised to become a fundamental part of the AI ecosystem. They leverage the raw power of models like LLMs and structure it into intelligent "workers" that can perceive, think, and act. As research and industry continue to develop this paradigm – making agents more reliable, aligned, and efficient (perhaps with many small models working in concert) – we can expect to see increasingly **sophisticated autonomous AI agents** tackling problems in education, healthcare, research, and virtually every domain of society. [19] [53]

---

[1] [12] [13] [15] [16] [20] [21] [22] [23] [25] [49] [50] [51] [53] [54] What is Agentic AI? - Agentic AI Explained - AWS
https://aws.amazon.com/what-is/agentic-ai/

[2] [3] [4] [5] [10] [11] [14] [17] [18] [19] [24] [48] [52] What is Agentic AI? | IBM
https://www.ibm.com/think/topics/agentic-ai

[6] [26] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [2506.02153] Small Language Models are the Future of Agentic AI
https://ar5iv.labs.arxiv.org/html/2506.02153

[7] [8] [9] AI - Agents & Environments
https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_agents_and_environments.htm

[27] [28] [29] [31] Is ChatGPT an AI Agent? | Understanding Its Tech Role
https://www.instituteofaistudies.com/insights/is-chatgpt-an-ai-agent

[30] Large language models revolutionized AI. LLM agents are what's next
https://research.ibm.com/blog/what-are-ai-agents-llm

[32] [2506.02153] Small Language Models are the Future of Agentic AI
https://arxiv.org/abs/2506.02153

[45] [46] [47] LLM Agents for Education: Advances and Applications
https://arxiv.org/html/2503.11733v1

[55] Building an Agentic Retrieval-Augmented Generation (RAG) system ...
https://www.facebook.com/AnalyticsVidhya/posts/building-an-agentic-retrieval-augmented-generation-rag-system-with-smolagents-en/1149931063804909/