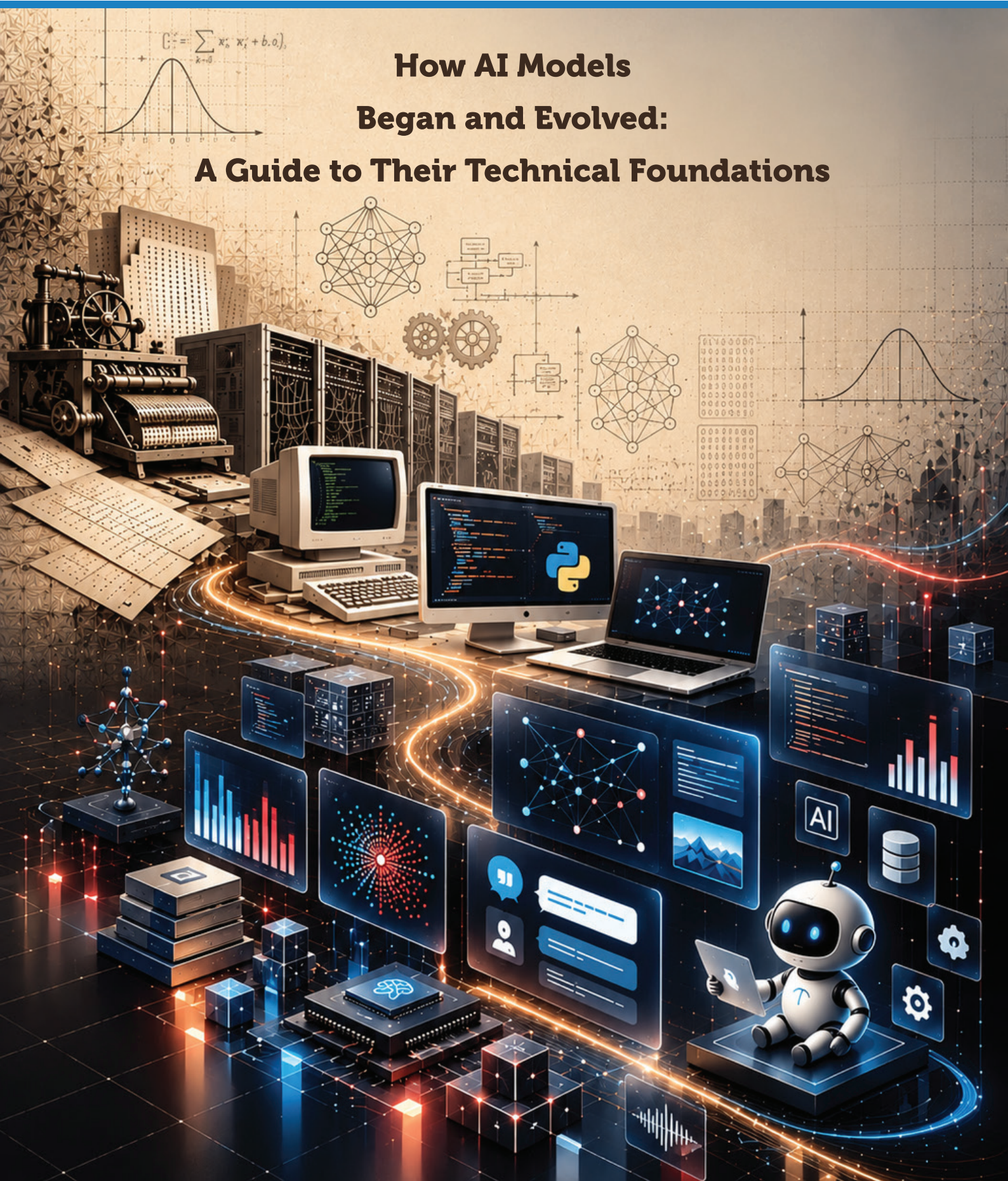




# How AI Models Began and Evolved: A Guide to Their Technical Foundations



Guide:

**How AI Models Began and Evolved:  
A Guide to Their Technical Foundations**

*San Marino, 27 October 2026*

*Disclaimer: This document is prepared by the researchers of the Centre for Global Studies (CGS) of the Parliamentary Assembly of the Mediterranean (PAM) in San Marino in their personal capacity. The opinions expressed in the note are the authors' own and may not reflect the views of PAM. This report underwent AI-assisted linguistic review.*

**Index**

|                                                        |    |
|--------------------------------------------------------|----|
| Executive Summary .....                                | 2  |
| Introduction.....                                      | 3  |
| How AI Models Began.....                               | 5  |
| From the Perceptron to Language Models.....            | 6  |
| Understanding the Large Language Model.....            | 7  |
| From Language Models to Conversational Assistants..... | 10 |
| From Conversational Assistants to AI Agents .....      | 12 |
| How AI Participates in Developing AI .....             | 15 |
| Conclusion .....                                       | 18 |

## Executive Summary

This guide, *How AI Models Began and Evolved: A Guide to Their Technical Foundations*, has been prepared by the Centre for Global Studies of the Parliamentary Assembly of the Mediterranean (PAM-CGS). It examines how artificial intelligence (AI) capabilities have expanded over decades through advances in learning methods, language processing, model architectures, computing resources, and supporting software. Particular attention is given to the distinction between capabilities acquired during training and the functions enabled by the systems in which AI models operate.

The guide explains how large language models (LLMs) learn relationships within text by adjusting numerical values during training. Their training material is selected by their developers; there is no single original database shared by all models. Text-generating models use these learned values to predict subsequent units of text, producing responses through repeated predictions. The introduction of the Transformer architecture in 2017 improved the handling of relationships within language and supported more efficient training at larger scales. Additional training using response examples and feedback can improve instruction-following behavior. During use, prompts direct existing capabilities, while searches and document retrieval supply information beyond the training material. A system can therefore answer using recently obtained information without incorporating that information into the model's trained parameters. Access to information and learning through training are distinct processes.

AI agents extend these capabilities through tools and software that coordinate continuing sequences of operations. Software executes requests generated by the model and returns results that inform subsequent steps. Task records and coordination between several agents can support more extensive work. The degree of operational independence depends on the system's design, available tools, permissions, and stopping conditions.

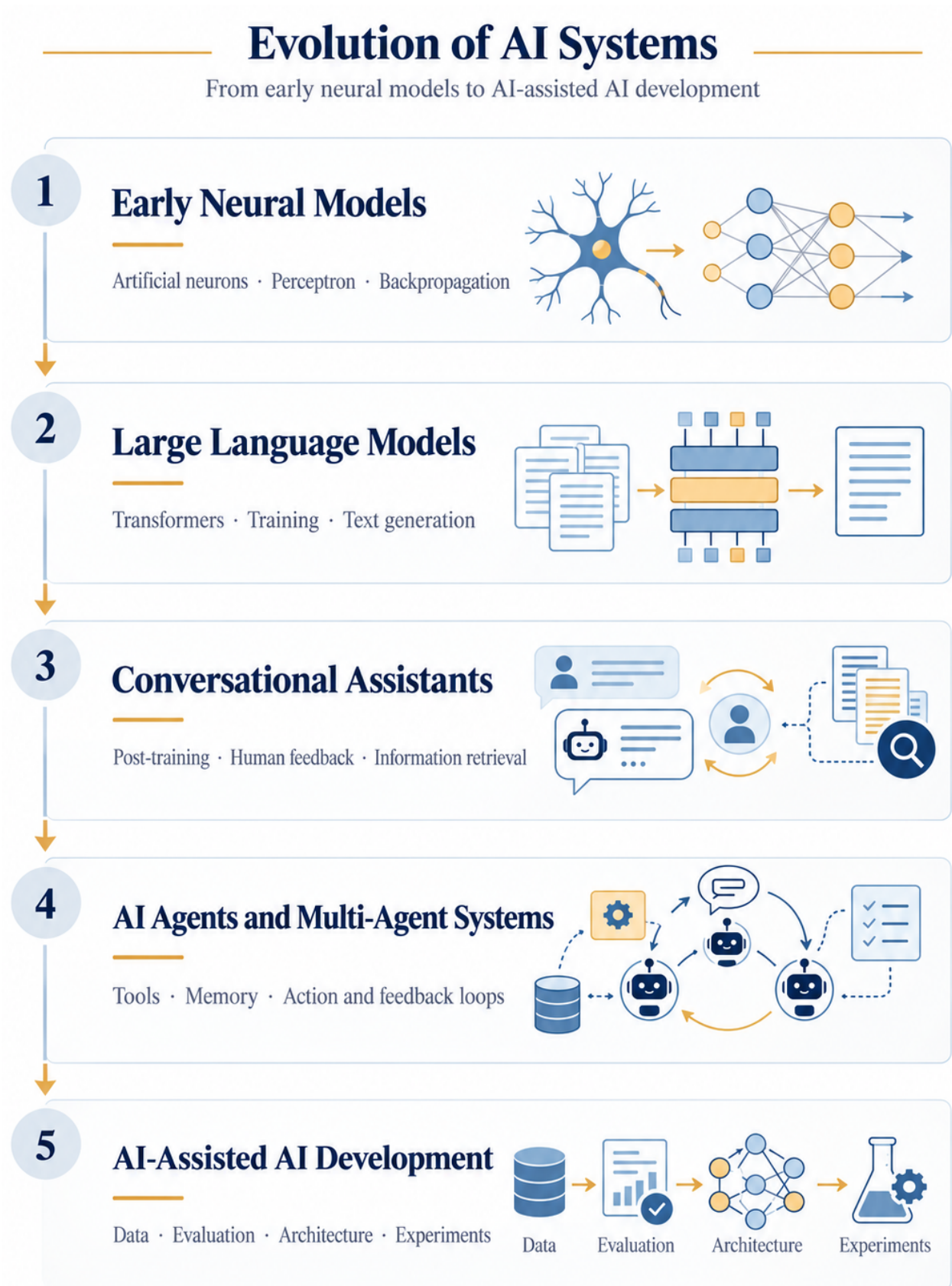
AI contributes to developing other AI systems by preparing training material, supplying feedback, testing model designs and learning procedures, improving computational efficiency, and conducting experiments. Experimental systems have also improved the software coordinating their own research activity while keeping the language models directing that activity fixed. These results demonstrate improvements to specific components and processes; they do not, by themselves, establish fully autonomous development of successor models.

The significance of a reported advance depends on what changed, how the improvement was evaluated, and the conditions under which the system operates. Distinguishing these factors provides a more precise basis for assessing AI capabilities, autonomy, and potential risks, including the activities and components addressed by legislative and regulatory proposals.

## **Introduction**

1. This guide, *How AI Models Began and Evolved: A Guide to Their Technical Foundations*, has been prepared by the Centre for Global Studies of the Parliamentary Assembly of the Mediterranean (PAM-CGS). It provides parliamentarians, diplomats, and policymakers with an accessible account of the technical processes through which contemporary artificial intelligence (AI) models and systems have developed.
2. The guide traces the progression from early artificial neural models to large language models, conversational assistants, and AI agents. It explains how models learn from data, how their design and computing resources shape their development, and how trained models obtain additional information and interact with tools. It also examines how AI systems participate in developing other AI systems, identifying the components and processes they help improve.
3. Understanding these foundations supports a more precise assessment of AI capabilities, their applications, and the conditions under which risks arise. As existing models become available for adaptation and integration into different systems, informed policy discussions require an understanding of how those systems are constructed and operated. This guide provides technical context for parliamentary deliberation, diplomatic dialogue, and consideration of legislative and regulatory proposals. It complements PAM-CGS's wider analysis of AI's contribution to development, its misuse, and emerging risks.

**Figure 1: Evolution of AI Systems<sup>1</sup>**



<sup>1</sup> Graphic compiled by the CGS research team, based on the analysis presented in this report.

## How AI Models Began

4. One foundational path to today's AI models began with efforts to represent the activity of neurons mathematically. In 1943, a framework based on simplified artificial neurons was presented in *A logical calculus of the ideas immanent in nervous activity*. Each unit received signals and produced an output when defined conditions were met. Connecting these units enabled a network to perform logical operations. Its connections and activation rules were specified in advance; the framework did not include a procedure for learning them from examples<sup>2</sup>.
5. The broader ambition took shape in the 1955 proposal for the Dartmouth research project, planned for 1956. Its authors used the term “artificial intelligence” and proposed investigating how machines could use language, form concepts, solve problems, and improve themselves<sup>3</sup>.
6. A significant advance toward systems capable of learning came with the perceptron, developed at the Cornell Aeronautical Laboratory and demonstrated publicly in 1958. Initially simulated on an IBM computer, the system was designed to recognize patterns. Its learning procedure adjusted its responses using supplied examples, reducing the need to specify every classification rule manually. The training material consisted of controlled visual patterns, establishing an early basis for learning from data well before the extensive text collections used by contemporary language models<sup>4</sup>.
7. Learning depended on converting information into numerical inputs whose influence on the output was governed by adjustable values called “weights.” During supervised training, examples were paired with their correct classifications. The system calculated an output, compared it with the expected result, and adjusted its weights when an error occurred. Repeated adjustments produced a numerical configuration capable of distinguishing particular categories of input. The design, training examples, and adjustment rule were specified in advance; the resulting weights were determined through training<sup>5</sup>.
8. These early systems had limited capabilities, but they established an important principle: information could shape a model's behavior through changes to its internal connections.

---

<sup>2</sup> Warren S. McCulloch and Walter Pitts. (December 1943). *A logical calculus of the ideas immanent in nervous activity*. *Bulletin of Mathematical Biophysics*. <https://doi.org/10.1007/BF02478259>

<sup>3</sup> John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. (31 August 1955). *A proposal for the Dartmouth summer research project on artificial intelligence*. Stanford University, archival reproduction. <https://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>

<sup>4</sup> Melanie Lefkowitz. (25 September 2019). *Professor's perceptron paved the way for AI – 60 years too soon*. Cornell Chronicle. <https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon>

<sup>5</sup> Cornell University. (n.d.). *Lecture 3: The Perceptron*. <https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html>

Extending that principle to more complex networks and finding effective ways to train them became central to the development of modern neural models.

### **From the Perceptron to Language Models**

9. The early perceptron demonstrated that a machine could learn from examples by adjusting its internal connections. Extending this principle to more complex tasks required changes in both the structure of the network and the method used to train it. The progression toward language models depended on making those connections capable of capturing increasingly complex relationships in data. An artificial neural network consists of connected processing units that perform calculations on numerical inputs. These units can be arranged in layers, with each layer carrying out a stage of processing. The output of one layer becomes input for the next, allowing the network to process information through successive stages<sup>6</sup>.
10. Training networks with several layers required a way to determine how an error at the output should affect connections within the network. In 1986, the use of backpropagation to address this problem was demonstrated in Learning representations by back-propagating errors<sup>7</sup>. The procedure worked backward through the network to calculate how changes to individual weights would affect the output error. These calculations guided adjustments across the layers. Through repeated training, intermediate units could learn features useful for the task without requiring each feature to be specified in advance.
11. Applying neural networks to language required words to be represented in a form that could be processed mathematically. In 2003, Université de Montréal demonstrated how this could be achieved in A Neural Probabilistic Language Model. The approach enabled a system to learn numerical values representing each word's use in the training text. Words with related linguistic roles could acquire related representations. This allowed relationships learned from one sequence to inform the model's assessment of other sequences<sup>8</sup>.
12. The system learned these word representations together with the calculations used to predict the next word from the preceding context. During training, the text supplied the actual continuation against which predictions could be assessed. Adjusting the model improved its estimates of which words were likely to follow particular combinations. This explains the term "language model": a trained mathematical system that estimates

---

<sup>6</sup> Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. (27 May 2015). Deep learning. *Nature*. <https://doi.org/10.1038/nature14539>

<sup>7</sup> David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. (9 October 1986). Learning representations by back-propagating errors. *Nature*. <https://doi.org/10.1038/323533a0>

<sup>8</sup> Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (February 2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3, 1137–1155. <https://www.jmlr.org/papers/v3/bengio03a.html>

relationships and probabilities within language. Its usefulness came partly from applying those learned relationships to word combinations absent from its training material<sup>9</sup>.

13. The development of language models advanced in 2017, when Google Brain, Google Research, and the University of Toronto introduced the Transformer architecture in Attention Is All You Need. Earlier recurrent networks commonly processed sequences through successive steps, carrying information from one position to the next. The Transformer introduced an architecture in which attention mechanisms calculated relationships directly between different positions in a sequence. This provided a different way to process the connections within language and established the basis for the developments examined below<sup>10</sup>.
14. “Attention” here describes a mathematical operation that assigns different degrees of influence to information elsewhere in the available context. It allows the representation at one position to incorporate relevant information from other positions. The architecture also enabled more calculations to run simultaneously during training, reducing the sequential processing constraint. These changes improved both the handling of relationships within language and the efficiency of training, providing an important foundation for the development of larger language models<sup>11</sup>.

### **Understanding the Large Language Model**

15. The Transformer introduced in the previous chapter provided a more effective way to process relationships within text. Building on this architecture, language models developed into systems capable of generating extended passages. Understanding a large language model, or LLM<sup>12</sup>, therefore requires understanding how a network turns language into numerical representations, learns from them, and uses what it has learned to produce new text<sup>13</sup>.
16. The “model” is the computational system formed by a network’s design and its trained numerical parameters. The design determines how information passes through its layers; the parameters determine how calculations within those layers operate. These parameters include weights that strengthen, weaken, or combine different signals. Their values are

---

<sup>9</sup> Ibidem

<sup>10</sup> Ashish Vaswani et al. (12 June 2017). Attention Is All You Need. arXiv. <https://arxiv.org/abs/1706.03762>

<sup>11</sup> Ibidem

<sup>12</sup> Zhao, W. X., et al. (9 May 2026). A Survey of Large Language Models. Frontiers of Computer Science, 20, Article 2012627. Springer Nature. <https://doi.org/10.1007/s11704-026-60308-3>

<sup>13</sup> Google. (2 January 2026). LLMs: What's a large language model? Google Developers. <https://developers.google.com/machine-learning/crash-course/llm/transformers>

adjusted during training and retained afterward. Together, they enable the network to recognize patterns and produce predictions when it encounters new input<sup>14</sup>.

17. Language must first be converted into a form this system can process. A tokenizer divides text into units called tokens<sup>15</sup>, which can represent words, parts of words, or punctuation. Each token receives a numerical identifier. The network then associates it with an embedding: a list of numerical values learned during training. Information about token positions is also incorporated. This gives the system numerical representations of both the text's components and their order<sup>16</sup>.
18. These representations pass through successive Transformer layers. Within each layer, attention calculates how strongly different tokens should influence one another, while other operations transform the resulting information. A token's representation therefore changes according to its context. Across the layers, the network can combine relationships involving nearby words, distant references, and the wider passage. Training shapes the parameters used in these calculations, allowing the network to learn relationships rather than requiring developers to specify each relationship individually.
19. The material used for this learning is a collection of texts, commonly called a training corpus. Its composition varies between models; there is no single original database shared by all LLMs. OpenAI, for example, identifies publicly available information, material accessed through partnerships, and information supplied or generated by users and people contributing to model development as principal sources. Preparing training material involves selecting and filtering content. These texts provide the examples from which the model's internal parameters are developed<sup>17</sup>.
20. For a model trained to generate text, a central training task is predicting the next token from the preceding tokens. Initially, its untrained parameters produce poor predictions. The training process compares the probabilities it assigns with the token that actually follows in the text. Backpropagation, explained in the previous chapter, determines adjustments that would reduce this error. Repeating this process across extensive text changes the

---

<sup>14</sup> OpenAI. (n.d.). How ChatGPT and our foundation models are developed. <https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-foundation-models-are-developed>; Vaswani, A., et al. (12 June 2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30 (NeurIPS 2017). <https://arxiv.org/abs/1706.03762>

<sup>15</sup> Kudo, T., & Richardson, J. (November 2018). SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. Association for Computational Linguistics. <https://aclanthology.org/D18-2012/>

<sup>16</sup> Google. (2 January 2026). LLMs: What's a large language model? Google Developers. <https://developers.google.com/machine-learning/crash-course/llm/transformers>; Vaswani, A., et al. (12 June 2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30 (NeurIPS 2017). <https://arxiv.org/abs/1706.03762>

<sup>17</sup> OpenAI. (n.d.). How ChatGPT and our foundation models are developed. <https://help.openai.com/en/articles/7842364-how-chatgpt-and-our-foundation-models-are-developed>

parameters throughout the network. Because the text supplies its own target tokens, this stage does not require a person to write a correct answer for every training example. This initial learning stage is called pretraining<sup>18</sup>.

21. It is a “language model” because it learns to estimate how language is structured: which tokens are likely in a given context and how their combinations form meaningful sequences. In the text-generating models described here, its immediate prediction is a set of probabilities for the next token. Producing these probabilities draws on relationships learned across training material, including grammar, recurring factual associations, and patterns of explanation. The prediction concerns the continuation of text; it is not inherently a forecast of a real-world event.
22. During use, the user’s input supplies the starting context. The model calculates next-token probabilities, and the generation procedure selects a token, sometimes by sampling among possible continuations. That token is added to the context, and the process repeats. The resulting token sequence is converted back into readable text. Sentences and paragraphs emerge through these successive steps, with each continuation influenced by the prompt and the text already generated. This use of trained parameters is called inference; ordinary text generation does not itself update those parameters<sup>19</sup>.
23. “Large” generally refers to the number of trainable parameters, which can reach billions. A larger network provides greater capacity to represent complex relationships, but effective expansion also depends on sufficient training data and computing resources. Computing hardware performs the calculations required to adjust the parameters. In 2022, DeepMind examined this relationship in Training Compute-Optimal Large Language Models, demonstrating that model size and the amount of training text needed to be balanced within the available computing budget. Increasing the number of parameters alone was therefore insufficient to make the most effective use of those resources<sup>20</sup>.
24. The development of the GPT family illustrates how these elements came together. In 2018, OpenAI presented the model now commonly called GPT-1 and documented its training approach in Improving Language Understanding by Generative Pre-Training. A Transformer network was first trained on a collection of books to predict subsequent

---

<sup>18</sup> Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI. [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf); Alec Radford. (11 June 2018). Improving language understanding with unsupervised learning. OpenAI. <https://openai.com/index/language-unsupervised/>

<sup>19</sup> Tom B. Brown et al. (28 May 2020). Language Models are Few-Shot Learners. arXiv. <https://arxiv.org/abs/2005.14165>

<sup>20</sup> Jordan Hoffmann et al. (29 March 2022). Training Compute-Optimal Large Language Models. arXiv. <https://arxiv.org/abs/2203.15556>

tokens, then adapted to specific language tasks. The name GPT captures this approach: “Generative” refers to producing language, “Pre-trained” to the initial learning stage, and “Transformer” to the network architecture<sup>21</sup>.

25. In 2020, OpenAI expanded this approach with GPT-3, presented in Language Models are Few-Shot Learners. Its largest version contained 175 billion parameters and was trained on material including filtered web pages, books, and Wikipedia. The results demonstrated that expansion improved the model’s ability to perform different language tasks using instructions or examples supplied in its input, without updating its parameters for each task. This progression combined larger networks, broader training material, and greater computing resources with the established process of learning from text and generating successive tokens<sup>22</sup>.

### **From Language Models to Conversational Assistants**

26. The language models described in the previous chapter could generate text and respond to instructions supplied in their input. The next step was to make this behavior more dependable. Additional training after pretraining, broadly called post-training, refined how the model responded to requests<sup>23</sup>.
27. One method is supervised fine-tuning. Human trainers provide instructions paired with examples of suitable responses. Training adjusts the model’s parameters to increase the likelihood of producing those responses. The model retains its previously developed language capabilities while learning more specific patterns of behavior, including how to answer a request in the expected form.
28. A further method uses comparisons between alternative responses. Human evaluators rank the outputs, and a separate reward model learns to estimate which responses they would prefer. Reinforcement learning then adjusts the language model’s parameters to favor responses receiving higher scores. This procedure, called reinforcement learning from human feedback, or RLHF, translates human evaluations into a numerical signal that can guide further training<sup>24</sup>.
29. In 2022, OpenAI demonstrated how human feedback could refine instruction-following behavior through the InstructGPT models, as described in Training language models to

---

<sup>21</sup> OpenAI. (11 June 2018). Improving language understanding with unsupervised learning.

<https://openai.com/index/language-unsupervised/>

<sup>22</sup> Tom B. Brown et al. (28 May 2020). Language Models are Few-Shot Learners. arXiv. <https://arxiv.org/abs/2005.14165>

<sup>23</sup> Ouyang, L., et al. (2022). Training Language Models to Follow Instructions with Human Feedback. Advances in Neural Information Processing Systems, 35 (NeurIPS 2022), 27730–27744. <https://arxiv.org/abs/2203.02155>

<sup>24</sup> Ibidem

follow instructions with human feedback. OpenAI subsequently introduced ChatGPT on 30 November 2022. Its initial version was refined from a model in the GPT-3.5 series using demonstrations of conversations and rankings of alternative responses. Human trainers contributed exchanges in which they played both the user and the assistant, helping shape the model's conversational behavior<sup>25</sup>.

30. This development process differs from giving instructions during a conversation. A prompt supplies information that the trained model uses when generating its response. Instructions, examples, or corrections can change the answer by changing that input, while the model's parameters remain unchanged. Fine-tuning changes the numerical values through additional training; prompting directs how the existing values are used. A model can therefore adjust its response to new information in a conversation without undergoing a new training process<sup>26</sup>.
31. This distinction also explains how a system can answer questions about information published after its training. The information learned during training reflects the material available at that stage. Developers can incorporate additional material through further training, but an assistant can also receive fresh information when a question is asked. A document supplied by the user or material retrieved from an external source becomes part of the context available for generating the answer<sup>27</sup>.
32. Web search provides one route to this external information. When search is enabled, the system can issue a query, obtain results, and access relevant content before producing its response. The model uses the retrieved material alongside the user's request. Source links can accompany the answer because the search process supplies identifiable pages and their associated information. OpenAI's web-search documentation describes this connection between access to current material and responses containing source citations<sup>28</sup>.
33. A related approach retrieves information from a designated collection of documents or databases. This is called retrieval-augmented generation, or RAG. In one common implementation, documents are divided into passages, converted into numerical representations called embeddings, and indexed for search. The question is also represented numerically, allowing the retrieval system to identify passages with related meaning, even

---

<sup>25</sup> OpenAI. (30 November 2022). Introducing ChatGPT. <https://openai.com/index/chatgpt/>

<sup>26</sup> Google. (3 December 2025). LLMs: Fine-tuning, distillation, and prompt engineering. Google Developers. <https://developers.google.com/machine-learning/crash-course/llm/tuning>

<sup>27</sup> Ibidem

<sup>28</sup> OpenAI. (n.d.). Web search. <https://developers.openai.com/api/docs/guides/tools-web-search>

when their wording differs. Relevant passages are then supplied to the language model, which uses them to formulate its answer<sup>29</sup>.

34. The searchable collection and the language model's trained parameters serve different functions. Updating the collection can make new material available for subsequent answers without retraining the language model. The assistant thus combines capabilities developed through training with information supplied during use. Its access to a recent document reflects a connection between the model and an external information system; incorporating that document into the model's trained parameters would require a separate training process.
35. OpenAI has been used here as an illustrative example. The principles highlighted in this chapter, including pretraining, refining model behavior, and supplying information during use, are broadly shared across major language-model developers, including OpenAI<sup>30</sup>, Anthropic<sup>31</sup>, Meta<sup>32</sup>, SpaceX Super Intelligence (formerly xAI)<sup>33</sup>, Google DeepMind<sup>34</sup>, and Mistral AI<sup>35</sup>, as well as Chinese developers such as DeepSeek<sup>36</sup> and Alibaba Cloud<sup>37</sup>. These principles also apply to online AI tools built around language models, whether they use open-source models, proprietary models supplied by other developers, or models developed in-house<sup>38</sup>. Specific training methods and system configurations vary across these providers and tools. Connecting these models to tools that perform operations and enabling them to select successive actions in response to results provides the foundation for AI agents, the next stage examined in this booklet.

## From Conversational Assistants to AI Agents

36. The previous chapter explained how a conversational assistant can follow instructions and obtain information from external sources. A further development connected these capabilities to a process that could continue working toward a goal. In this setting, an AI

---

<sup>29</sup> Google Cloud. (n.d.). What is Retrieval-Augmented Generation (RAG)? <https://cloud.google.com/use-cases/retrieval-augmented-generation>; OpenAI. (n.d.). Retrieval. <https://developers.openai.com/api/docs/guides/retrieval>

<sup>30</sup> OpenAI: Long Ouyang et al. (4 March 2022). Training language models to follow instructions with human feedback. arXiv. <https://arxiv.org/abs/2203.02155>

<sup>31</sup> Anthropic. (15 December 2022). Constitutional AI: Harmlessness from AI feedback. <https://www.anthropic.com/research/constitutional-ai-harmlessness-from-ai-feedback>

<sup>32</sup> Meta. (23 July 2024). Introducing Llama 3.1: Our most capable models to date. <https://ai.meta.com/blog/meta-llama-3-1/>

<sup>33</sup> SpaceX Super Intelligence. (n.d.). Get started with SpaceX Super Intelligence. <https://docs.x.ai/overview>

<sup>34</sup> Google DeepMind: Gemini Team, Google. (8 March 2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv. <https://arxiv.org/abs/2403.05530>

<sup>35</sup> Mistral AI: Mistral team. (26 February 2024). Au Large. Mistral AI. <https://mistral.ai/news/mistral-large/>

<sup>36</sup> DeepSeek: DeepSeek-AI et al. (27 December 2024). DeepSeek-V3 Technical Report. arXiv. <https://arxiv.org/abs/2412.19437>

<sup>37</sup> Alibaba Cloud: Qwen Team. (29 April 2025). Qwen3: Think Deeper, Act Faster. <https://qwenlm.github.io/blog/qwen3/>

<sup>38</sup> Artificial Analysis. (n.d.). Models. <https://artificialanalysis.ai/models/releases>

agent is a system built around a language model, with tools for performing operations and software for managing successive steps. The model helps determine what to do next, while the surrounding system enables the operation and returns its result<sup>39</sup>.

37. This connection begins with descriptions of the tools available to the model: what each tool does and what information it requires. Using the task instructions and these descriptions, the model can generate a structured request identifying a tool and its inputs. This request is another form of model output, produced through the language-processing mechanism described earlier. Software interprets the request and executes the corresponding operation. The connection therefore turns a generated instruction into an action performed by another component of the system<sup>40</sup>.
38. The operation's result is then supplied to the model as additional input. The model can assess what happened and generate a further request, revise its approach, or produce a final response. Repeating this sequence creates an agent loop: select an operation, execute it, receive the result, and determine the next step. The software managing this loop coordinates the exchange between the model and its tools and applies the system's stopping conditions. Each step can therefore depend on information obtained during the preceding step<sup>41</sup>.
39. Continuing across many steps also requires maintaining a record of the task. Instructions, plans, completed operations, and relevant results can be retained in the model's context, the information available for its current computation. Because the context window limits how much information can be processed at once, longer tasks may require summaries or external storage. Information stored outside the model can subsequently be retrieved and supplied again. This working memory helps preserve continuity across successive operations<sup>42</sup>.
40. The degree of control given to the model distinguishes different system designs. In a fixed workflow, software determines the sequence of operations in advance. In an agent, the model has greater scope to select which operation should follow, using the goal and the results obtained so far. Developers still define the available tools and the conditions under

---

<sup>39</sup> Erik S. and Barry Zhang. (19 December 2024). *Building effective agents*. Anthropic. <https://www.anthropic.com/engineering/building-effective-agents>

<sup>40</sup> OpenAI. (n.d.). Function calling. <https://developers.openai.com/api/docs/guides/function-calling>; Anthropic. (n.d.). Tool use with Claude. <https://platform.claude.com/docs/en/agents-and-tools/tool-use/overview>

<sup>41</sup> Ibidem

<sup>42</sup> Wu, Q., et al. (2024). AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. Proceedings of the First Conference on Language Modeling (COLM 2024). Microsoft Research. <https://www.microsoft.com/en-us/research/publication/autogen-enabling-next-gen-llm-applications-via-multi-agent-conversation-framework/>

which the system operates. Agentic behavior therefore depends on both the model’s capabilities and the design of the process surrounding it<sup>43</sup>.

41. In 2022, Princeton University and Google Research demonstrated how language models could combine reasoning with action in ReAct: Synergizing Reasoning and Acting in Language Models. The approach interleaved reasoning expressed in text with operations that obtained external information or interacted with an environment. The resulting observations informed subsequent reasoning, allowing the model to revise its plan and respond to unexpected results. This established an influential basis for systems in which reasoning, action, and feedback form a continuing process<sup>44</sup>.
42. Subsequent developments made these connections more practical. On 13 June 2023, OpenAI introduced models fine-tuned to identify when a function should be called and generate the structured inputs required for that call<sup>45</sup>. On 22 October 2024, Anthropic introduced computer use for Claude 3.5 Sonnet. Through an appropriate software setup, the model could interpret screenshots and request cursor movements, clicks, and keyboard input. The same action-and-feedback process could consequently operate through a computer’s visual interface<sup>46</sup>.
43. The architecture can also coordinate several agents. A lead agent can divide a task, assign parts to other agents, and combine their findings. Each agent operates with its own instructions, context, and available tools. In 2025, Anthropic described such an arrangement in How we built our multi-agent research system: a lead agent developed a strategy, delegated investigations, and used the returned findings to decide whether further work was needed. The system expanded its capacity through coordinated model interactions and separate working contexts<sup>47</sup>.
44. Revising a plan, retaining a result, or selecting a different tool can change an agent’s behavior during a task while the underlying model’s trained parameters remain unchanged. These adjustments occur through the information and software used around the model. Changing the model’s learned parameters requires a training process<sup>48</sup>. This distinction

---

<sup>43</sup> Erik S. and Barry Zhang. (19 December 2024). Building effective agents. Anthropic.

<https://www.anthropic.com/engineering/building-effective-agents>

<sup>44</sup> Shunyu Yao et al. (6 October 2022). ReAct: Synergizing Reasoning and Acting in Language Models. arXiv. <https://arxiv.org/abs/2210.03629>

<sup>45</sup> OpenAI. (13 June 2023). Function calling and other API updates. <https://openai.com/index/function-calling-and-other-api-updates/>

<sup>46</sup> Anthropic. (22 October 2024). Developing a computer use model. <https://www.anthropic.com/news/developing-computer-use>

<sup>47</sup> Jeremy Hadfield, Barry Zhang, Kenneth Lien, Florian Scholz, Jeremy Fox, and Daniel Ford. (13 June 2025). How we built our multi-agent research system. Anthropic. <https://www.anthropic.com/engineering/multi-agent-research-system>

<sup>48</sup> Google. (3 December 2025). LLMs: Fine-tuning, distillation, and prompt engineering. Google Developers. <https://developers.google.com/machine-learning/crash-course/llm/tuning>

provides the starting point for the next chapter: how AI systems participate in developing other AI systems, and which parts of that development process they help change.

### **How AI Participates in Developing AI**

45. The previous chapter explained how a language model connected to tools can operate as an agent. Those tools can also give it access to programming environments, training systems, and experimental results. AI can therefore participate in the process through which other AI systems are developed. Its contributions extend from preparing training material to testing changes in models and improving the software used to build them.
46. One contribution begins with the data. Existing models can classify documents by quality, select useful material, and generate additional training text. This generated material is called synthetic data. NVIDIA's Nemotron-CC project, published in 2024, combined original web material with model-generated text and used model-based classifiers to improve selection. The resulting material becomes part of a training dataset: its influence enters another model through the adjustments made to that model's parameters during training. An improved model can subsequently help prepare higher-quality training material for the next training cycle<sup>49</sup>.
47. Another contribution involves a model acting as a teacher. In knowledge distillation, a student model is trained using a teacher's responses or the probabilities it assigns to possible outputs. The student's own parameters are adjusted to learn patterns reflected in those outputs. This can transfer capabilities into a smaller model that requires fewer computing resources<sup>50</sup>. DeepSeek demonstrated this approach in 2025 by using examples generated through DeepSeek-R1 to refine smaller models based on Alibaba's Qwen and Meta's Llama<sup>51</sup>. The transfer occurred through additional training of those models.
48. AI can also provide the evaluations that guide training. The previous chapters described how human rankings help shape a model's responses; some methods obtain such comparisons from another model. Anthropic's Constitutional AI, presented in 2022, used human-written principles to guide model-generated critiques and revised answers. Those revisions supplied examples for fine-tuning. AI comparisons between alternative answers also trained a preference model, whose scores guided reinforcement learning. AI thus

---

<sup>49</sup> Dan Su et al. (3 December 2024). Nemotron-CC: Transforming Common Crawl into a Refined Long-Horizon Pretraining Dataset. NVIDIA. <https://research.nvidia.com/labs/adlr/Nemotron-CC/>

<sup>50</sup> Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. (9 March 2015). Distilling the Knowledge in a Neural Network. arXiv. <https://arxiv.org/abs/1503.02531>

<sup>51</sup> DeepSeek-AI et al. (22 January 2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv. <https://arxiv.org/abs/2501.12948>

helped produce both the examples to learn from and the numerical feedback used to adjust behavior<sup>52</sup>.

49. AI's contribution can extend to the design of a model's architecture: the arrangement of layers and connections through which calculations take place. In 2016, Google Brain demonstrated an automated approach to this task in Neural Architecture Search with Reinforcement Learning<sup>53</sup>. A controller network proposed candidate designs. Each candidate was then built, trained, and assessed using data reserved for evaluation. Its performance provided feedback that trained the controller to propose more promising designs. Through this process, AI helped determine the structure within which another model would learn.
50. AI can also contribute to developing the procedure through which models learn. In 2025, Google DeepMind reported the discovery of a reinforcement learning rule, DiscoRL, in Discovering state-of-the-art reinforcement learning algorithms<sup>54</sup>. The rule was developed through the experience of a population of agents. One network generated numerical reference values, called targets, to guide the agents' predictions and the way they selected actions. During training, the agents adjusted their parameters to bring their outputs closer to those values. Feedback from their subsequent performance helped improve the network generating the targets. This process enabled AI to search for more effective learning procedures, changing the method that guided adjustments to the agents' parameters.
51. A further contribution concerns the computing work required to train models. An AI coding agent can propose alternative implementations of a calculation, run them, check their correctness, and compare their speed. Promising versions can guide subsequent attempts. Google DeepMind's AlphaEvolve, announced in 2025, applied this process to algorithms used in AI infrastructure. Google DeepMind reported that an improvement to a mathematical routine reduced Gemini's overall training time by 1%. AlphaEvolve also contributed to data center scheduling and part of an AI chip's circuit design, improving resources used to develop and operate models<sup>55</sup>.

---

<sup>52</sup> Anthropic. (15 December 2022). Constitutional AI: Harmlessness from AI feedback.

<https://www.anthropic.com/research/constitutional-ai-harmlessness-from-ai-feedback>

<sup>53</sup> Barret Zoph and Quoc V. Le. (5 November 2016). Neural Architecture Search with Reinforcement Learning. arXiv.

<https://arxiv.org/abs/1611.01578>

<sup>54</sup> Junhyuk Oh et al. (22 October 2025). Discovering state-of-the-art reinforcement learning algorithms. Nature.

<https://www.nature.com/articles/s41586-025-09761-x>

<sup>55</sup> AlphaEvolve team. (14 May 2025). AlphaEvolve: A Gemini-powered coding agent for designing advanced algorithms. Google DeepMind. <https://deepmind.google/blog/alphaevolve-a-gemini-powered-coding-agent-for-designing-advanced-algorithms/>

52. These contributions can be combined within an automated research process. An agent proposes an experimental change, writes the necessary code, launches training, examines the results, and decides what to test next. Anthropic demonstrated such work in April 2026 using nine Claude-based agents investigating how stronger models could learn from weaker models’ supervision. They developed ideas, ran experiments, and shared findings and code. Their successes concerned defined experiments with smaller models; applying the best method to production-scale Claude Sonnet 4 produced no statistically significant improvement. The demonstration nevertheless showed agents carrying out successive stages of model-development research<sup>56</sup>.
53. The software coordinating a research agent can itself become the subject of improvement. In September 2026, Weco AI<sup>57</sup> described an automated approach to this task in its preprint Recursive self-improvement of AI research agents. Its AIDE<sup>2</sup> system proposed changes to an agent’s code, evaluated modified versions, and retained improvements. Each accepted version provided the starting point for subsequent changes. An eight-day run produced seven successive improvements, including changes to search strategies and memory management. The language models directing the loops remained fixed; the improvements affected the software organizing their research activity<sup>58</sup>.
54. By 2026, AI’s participation also extended beyond individual experiments into development work inside major laboratories. In its own assessment covering August 2026, Anthropic reported that Claude could lead 26% of its measured AI research and development work. “Lead” meant completing most of a task from a high-level prompt while a person supervised. The company reported no measured category operating fully autonomously. This provides a concrete picture of the current stage: AI systems can undertake substantial parts of developing their successors, while the extent of their independence varies across tasks and development stages<sup>59</sup>.

---

<sup>56</sup> Anthropic. (14 April 2026). Automated Alignment Researchers: Using large language models to scale scalable oversight. <https://www.anthropic.com/news/automated-alignment-researchers>

<sup>57</sup> Weco. (n.d.). Overview: Weco Autoresearch Engine. Weco Documentation. <https://docs.weco.ai/>

<sup>58</sup> Dhruv Srikanth, Bingchen Zhao, Dixing Xu, Yuxiang Wu, and Zhengyao Jiang. (22 September 2026). Recursive self-improvement of AI research agents. arXiv. <https://arxiv.org/abs/2609.26457>

<sup>59</sup> Anthropic. (n.d.). Measurements for understanding the pace of AI development inside frontier labs. <https://www.anthropic.com/institute/measuring-pace-of-ai-development>

## Conclusion

55. The development of AI reflects advances in data, model architecture, training, and the software through which models operate. These advances affect different components of a system. A model's learned capabilities, its access to external information, and its ability to act through connected tools must therefore be distinguished when assessing its performance. The same distinction helps clarify claims that AI is participating in its own development: improvements may concern the model, its training process, or the software coordinating its activity.
56. Understanding these components also provides a basis for examining where risks arise. A system's behavior depends on how its model has been trained or adapted, the information and tools available to it, and the conditions under which it operates. These considerations apply both to formally released systems and to systems modified or assembled for malicious purposes, including those distributed through the Dark Web. Technical understanding helps identify the capabilities and operating conditions underlying the risks examined in wider policy and security analysis.
57. Access to existing models and development tools has expanded opportunities to build customized AI systems. This includes adapting pretrained models and integrating them with additional software, information sources, and tools. For parliamentarians, diplomats, and policymakers, understanding these relationships supports a more precise assessment of the activities, actors, and components addressed by legislative and regulatory proposals. It also provides a basis for examining whether proposed measures correspond to the capabilities and uses they seek to govern.
58. These technical foundations complement PAM-CGS's continuing analysis of AI's contribution to development, its misuse, and emerging risks. Parliamentarians, diplomats, and policymakers may consult PAM-CGS publications<sup>60</sup> and its Daily and Weekly Digest on AI and Emerging Technologies for further analysis and updates<sup>61</sup>.

---

<sup>60</sup> PAM-CGS. (n.d.). Documents. <https://www.cgspam.org/category/ai-and-emerging-technologies/>

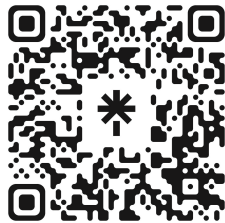
<sup>61</sup> PAM-CGS. (n.d.). Digest on AI and Emerging Technologies. <https://pam.int/digest-on-ai-and-emerging-technologies/>; <https://www.cgspam.org/digest/>

Prepared by  
the Parliamentary Assembly of the Mediterranean (PAM),  
Centre for Global Studies (CGS)

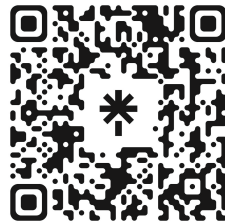
---

Viale Antonio Onofri, 47890 - 103  
Città di San Marino, San Marino

---



PAM



CGS

[www.pam.int](http://www.pam.int)  
[www.cgspam.org](http://www.cgspam.org)

