A **large language model** (**LLM**) is a type of [language model](https://en.wikipedia.org/wiki/Language_model) notable for its ability to achieve general-purpose language understanding and generation. LLMs acquire these abilities by using massive amounts of data to learn billions of parameters during training and consuming large computational resources during their training and operation.[[1]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-:7-1) LLMs are [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network) (mainly [transformers](https://en.wikipedia.org/wiki/Transformer_(machine_learning_model))[[2]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-2)) and are (pre-)trained using [self-supervised learning](https://en.wikipedia.org/wiki/Self-supervised_learning) and [semi-supervised learning](https://en.wikipedia.org/wiki/Semi-supervised_learning).

As autoregressive language models, they work by taking an input text and repeatedly predicting the next token or word.[[3]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Bowman-3) Up to 2020, [fine tuning](https://en.wikipedia.org/wiki/Fine-tuning_(machine_learning)) was the only way a model could be adapted to be able to accomplish specific tasks. Larger sized models, such as [GPT-3](https://en.wikipedia.org/wiki/GPT-3), however, can be [prompt-engineered](https://en.wikipedia.org/wiki/Prompt_engineering) to achieve similar results.[[4]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-few-shot-learners-4) They are thought to acquire knowledge about syntax, semantics and "ontology" inherent in human language corpora, but also inaccuracies and biases present in the corpora.[[5]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Manning-2022-5)

Notable examples include [OpenAI](https://en.wikipedia.org/wiki/OpenAI)'s [GPT](https://en.wikipedia.org/wiki/Generative_pre-trained_transformer) models (e.g., [GPT-3.5](https://en.wikipedia.org/wiki/GPT-3.5) and [GPT-4](https://en.wikipedia.org/wiki/GPT-4), used in [ChatGPT](https://en.wikipedia.org/wiki/ChatGPT)), [Google](https://en.wikipedia.org/wiki/Google)'s [PaLM](https://en.wikipedia.org/wiki/PaLM) (used in [Bard](https://en.wikipedia.org/wiki/Google_Bard)), and [Meta](https://en.wikipedia.org/wiki/Meta_Platforms)'s [LLaMa](https://en.wikipedia.org/wiki/LLaMA), as well as [BLOOM](https://en.wikipedia.org/wiki/BLOOM_(language_model)), [Ernie 3.0 Titan](https://en.wikipedia.org/wiki/Ernie_Bot), and [Anthropic's Claude 2](https://en.wikipedia.org/wiki/Anthropic#Claude).

## Dataset preprocessing[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=1)]

*See also:* [*List of datasets for machine-learning research § Internet*](https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research#Internet)

### **Probabilistic tokenization**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=2)]

Using a modification of [byte-pair encoding](https://en.wikipedia.org/wiki/Byte_pair_encoding), in the first step, all unique characters (including blanks and [punctuation marks](https://en.wikipedia.org/wiki/Punctuation_mark)) are treated as an initial set of [n-grams](https://en.wikipedia.org/wiki/N-grams) (i.e. initial set of uni-grams). Successively the most frequent pair of adjacent characters is merged into a bi-gram and all instances of the pair are replaced by it. All occurrences of adjacent pairs of (previously merged) n-grams that most frequently occur together are then again merged into even lengthier n-gram repeatedly until a vocabulary of prescribed size is obtained (in case of [GPT-3](https://en.wikipedia.org/wiki/GPT-3), the size is 50257).[[6]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-xbiWb-6) Token vocabulary consists of [integers](https://en.wikipedia.org/wiki/Integers), spanning from zero up to the size of the token vocabulary. New words can always be interpreted as combinations of the tokens and the initial-set uni-grams.[[7]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-2022Book_-7)

A token vocabulary based on the frequencies extracted from mainly English corpora uses as few tokens as possible for an average English word. An average word in another language encoded by such an English-optimized tokenizer is however split into suboptimal amount of tokens.

tokenizer: texts -> series of numerical "tokens" may be split into:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n-grams: | *token* | *izer* | ***:*** | *texts* | -> | *series* | *of* | *numerical* | " | *t* | *ok* | *ens* | " |
| numbers as "tokens": | 30001 | 7509 | 25 | 13399 | 4613 | 2168 | 286 | 29052 | 366 | 83 | 482 | 641 | 1 |

Probabilistic tokenization also [compresses](https://en.wikipedia.org/wiki/Data_compression) the datasets, which is the reason for using the [byte pair encoding](https://en.wikipedia.org/wiki/Byte_pair_encoding) algorithm as a tokenizer. Because LLMs generally require input to be an array that is not [jagged](https://en.wikipedia.org/wiki/Jagged_array), the shorter texts must be "padded" until they match the length of the longest one. How many tokens are, on average, needed per word depends on the language of the dataset.[[8]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-8)[[9]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-9)

### **Dataset cleaning**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=3)]

Removal of toxic passages from the dataset, discarding low-quality data, and de-duplication are examples of [dataset cleaning](https://en.wikipedia.org/wiki/Data_cleansing).[[10]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-aYNg4-10) Resulting, cleaned (high-quality) datasets contain up to 17 trillion words in 2022, raising from 985 million words, used in 2018 for [GPT-1](https://en.wikipedia.org/wiki/GPT-1),[[11]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-TzrRM-11) and 3.3 billion words, used for [BERT](https://en.wikipedia.org/wiki/BERT_(language_model)).[[12]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-jm-12) The future data is, however, expected to be increasingly "contaminated" by LLM-generated contents themselves.[[13]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-qbFw1-13)

## Training and architecture details[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=4)]

*See also:* [*Fine-tuning (machine learning)*](https://en.wikipedia.org/wiki/Fine-tuning_(machine_learning))

### **Reinforcement learning from human feedback (RLHF)**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=5)]

[Reinforcement learning from human feedback](https://en.wikipedia.org/wiki/Reinforcement_learning_from_human_feedback) (RLHF) through algorithms, such as [proximal policy optimization](https://en.wikipedia.org/wiki/Proximal_Policy_Optimization), is used to further fine-tune a model based on a dataset of human preferences.[[14]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-instructGPT-paper-14)

### **Instruction tuning**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=6)]

Using "self-instruct" approaches, LLMs have been able to [bootstrap](https://en.wikipedia.org/wiki/Bootstrapping) correct responses, replacing any naive responses, starting from human-generated corrections of a few cases. For example, in the instruction "Write an essay about the main themes represented in Hamlet," an initial naive completion might be 'If you submit the essay after March 17, your grade will be reduced by 10% for each day of delay," based on the frequency of this textual sequence in the corpus.[[15]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-self-instruct-paper-15)

### **Mixture of experts**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=7)]

*Main article:* [*Mixture of experts*](https://en.wikipedia.org/wiki/Mixture_of_experts)

The largest LLM may be too expensive to train and use directly. For such models, [mixture of experts](https://en.wikipedia.org/wiki/Mixture_of_experts) (MoE) can be applied, a line of research pursued by Google researchers since 2017 to train models reaching up to 1 billion parameters.[[16]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-HGZCJ-16)[[17]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-R9Qq5-17)[[18]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-glam-blog-18)

### **Prompt engineering, attention mechanism, and context window**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=8)]

*See also:* [*Prompt engineering*](https://en.wikipedia.org/wiki/Prompt_engineering) *and* [*Attention (machine learning)*](https://en.wikipedia.org/wiki/Attention_(machine_learning))

Most results previously achievable only by (costly) fine-tuning, can be achieved through [prompt engineering](https://en.wikipedia.org/wiki/Prompt_engineering), although limited to the scope of a single conversation (more precisely, limited to the scope of a context window).[[19]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-emergentpaper-19)

## Training cost[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=9)]

Advances in software and hardware have reduced the cost substantially since 2020, such that in 2023 training of a 12-billion-parameter LLM computational cost is 72,300 [A100-GPU](https://en.wikipedia.org/wiki/Ampere_(microarchitecture))-hours, while in 2020 the cost of training a 1.5-billion-parameter LLM (which was two orders of magnitude smaller than the state of the art in 2020) was between $80 thousand and $1.6 million.[[24]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Wiggers-24)[[25]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-xaytj-25)[[26]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Pythia-26) Since 2020, large sums were invested in increasingly large models. For example, training of the GPT-2 (i.e. a 1.5-billion-parameters model) in 2019 cost $50,000, while training of the PaLM (i.e. a 540-billion-parameters model) in 2022 cost $8 million.[[27]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-0BrVG-27)

For Transformer-based LLM, training cost is much higher than inference cost. It costs 6 [FLOPs](https://en.wikipedia.org/wiki/FLOPS) per parameter to train on one token, whereas it costs 1 to 2 FLOPs per parameter to infer on one token.[[28]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-kaplan-scaling-28)

## Tool use[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=10)]

There are certain tasks that, in principle, cannot be solved by any LLM, at least not without the use of external tools or additional software. An example of such a task is responding to the user's input '354 \* 139 = ', provided that the LLM has not already encountered a continuation of this calculation in its training corpus. In such cases, the LLM needs to resort to running program code that calculates the result, which can then be included in its response. Another example is 'What is the time now? It is ', where a separate program interpreter would need to execute a code to get system time on the computer, so LLM could include it in its reply.[[29]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-PI1fW-29)[[30]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-J5OW5-30) This basic strategy can be sophisticated with multiple attempts of generated programs, and other sampling strategies.[[31]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-gQxzq-31)

Generally, in order to get an LLM to use tools, one must finetune it for tool-use. If the number of tools is finite, then finetuning may be done just once. If the number of tools can grow arbitrarily, as with online [API](https://en.wikipedia.org/wiki/API) services, then the LLM can be finetuned to be able to read API documentation and call API correctly.[[32]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-lLrda-32)[[33]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-4Xzrs-33)

A simpler form of tool use is *Retrieval Augmented Generation*: augment an LLM with [document retrieval](https://en.wikipedia.org/wiki/Document_retrieval), sometimes using a [vector database](https://en.wikipedia.org/wiki/Vector_database). Given a query, a document retriever is called to retrieve the most relevant (usually measured by first encoding the query and the documents into vectors, then finding the documents with vectors closest in Euclidean norm to the query vector). The LLM then generates an output based on both the query and the retrieved documents.[[34]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-BUZBP-34)

## Agency[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=11)]

An LLM is a language model, which is not an agent as it has no goal, but it can be used as a component of an [intelligent agent](https://en.wikipedia.org/wiki/Intelligent_agent).[[35]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-CFuti-35) Researchers have described several methods for such integrations.

The ReAct ("Reason+Act") method constructs an [agent](https://en.wikipedia.org/wiki/Intelligent_agent) out of an LLM, using the LLM as a planner. The LLM is prompted to "think out loud". Specifically, the language model is prompted with a textual description of the environment, a goal, a list of possible actions, and a record of the actions and observations so far. It generates one or more thoughts before generating an action, which is then executed in the environment.[[36]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-DmvNE-36) The linguistic description of the environment given to the LLM planner can even be the LaTeX code of a paper describing the environment.[[37]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-JS8Vd-37)

In the DEPS ("Describe, Explain, Plan and Select") method, an LLM is first connected to the visual world via image descriptions, then it is prompted to produce plans for complex tasks and behaviors based on its pretrained knowledge and environmental feedback it receives.[[38]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-38)

The Reflexion method[[39]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-sbB2T-39) constructs an agent that learns over multiple episodes. At the end of each episode, the LLM is given the record of the episode, and prompted to think up "lessons learned", which would help it perform better at a subsequent episode. These "lessons learned" are given to the agent in the subsequent episodes.

[Monte Carlo tree search](https://en.wikipedia.org/wiki/Monte_Carlo_tree_search) can use an LLM as rollout heuristic. When a programmatic world model is not available, an LLM can also be prompted with a description of the environment to act as world model.[[40]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-ltTer-40)

For open-ended exploration, an LLM can be used to score observations for their "interestingness", which can be used as a reward signal to guide a normal (non-LLM) reinforcement learning agent.[[41]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-mBvD9-41) Alternatively, it can [propose increasingly difficult tasks](https://en.wikipedia.org/wiki/Zone_of_proximal_development) for [curriculum learning](https://en.wikipedia.org/w/index.php?title=Curriculum_learning&action=edit&redlink=1).[[42]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-:0-42) Instead of outputting individual actions, an LLM planner can also construct "skills", or [functions](https://en.wikipedia.org/wiki/Function_(computer_programming)) for complex action sequences. The skills can be stored and later invoked, allowing increasing levels of abstraction in planning.[[42]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-:0-42)

LLM-powered agents can keep a long-term memory of its previous contexts, and the memory can be retrieved in the same way as Retrieval Augmented Generation. Multiple such agents can interact socially.[[43]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-XuvjF-43)

## Compression[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=12)]

Typically, LLM are trained with full- or half-precision floating point numbers (float32 and float16). One float16 has 16 bits, or 2 bytes, and so one billion parameters require 2 gigabytes. The largest models typically have 100 billion parameters, requiring 200 gigabytes to load, which places them outside the range of most consumer electronics.

*Post-training* [*quantization*](https://en.wikipedia.org/wiki/Quantization_(signal_processing))[[44]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-LS2Go-44) aims to decrease the space requirement by lowering precision of the parameters of a trained model, while preserving most of its performance.[[45]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-cpzcK-45)[[46]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-QVU95-46) The simplest form of quantization simply truncates all numbers to a given number of bits. It can be improved by using a different quantization [codebook](https://en.wikipedia.org/wiki/Block_cipher) per layer. Further improvement can be done by applying [different precisions](https://en.wikipedia.org/wiki/Mixed-precision_arithmetic) to different parameters, with higher precision for particularly important parameters ("outlier weights").[[47]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-dU9Bu-47)

While quantized models are typically frozen, and only pre-quantized models are finetuned, quantized models can still be finetuned.[[48]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-D0nFA-48)

## Multimodality[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=13)]

Multimodality means "having several modalities", and a ["modality"](https://en.wikipedia.org/wiki/Modality_(human%E2%80%93computer_interaction)) means a type of input, such as video, image, audio, text, [proprioception](https://en.wikipedia.org/wiki/Proprioception), etc.[[49]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-49) There have been many AI models trained specifically to ingest one modality and output another modality, such as [AlexNet](https://en.wikipedia.org/wiki/AlexNet) for image to label,[[50]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-50) [visual question answering](https://en.wikipedia.org/wiki/Visual_question_answering) for image-text to text,[[51]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-51) and [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition) for speech to text. A review article of multimodal LLM is.[[52]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-52)

A common method to create multimodal models out of an LLM is to "tokenize" the output of a trained encoder. Concretely, one can construct a LLM that can understand images as follows: take a trained LLM, and take a trained image encoder *�*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{62},0.6666666666666666,0.6666666666666666. Make a small multilayered perceptron *�*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{63},0.6666666666666666,0.6666666666666666, so that for any image *�*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{64},0.6666666666666666,0.6666666666666666, the post-processed vector *�(�(�))*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{65},1.1875,0.6666666666666666 has the same dimensions as an encoded token. That is an "image token". Then, one can interleave text tokens and image tokens. The compound model is then finetuned on an image-text dataset. This basic construction can be applied with more sophistication to improve the model. The image encoder may be frozen to improve stability.[[53]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-53)

Flamingo demonstrated the effectiveness of the tokenization method, finetuning a pair of pretrained language model and image encoder to perform better on visual question answering than models trained from scratch.[[54]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-54) [Google PaLM](https://en.wikipedia.org/wiki/Pathways_Language_Model) model was finetuned into a multimodal model PaLM-E using the tokenization method, and applied to robotic control.[[55]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-55) [LLaMA](https://en.wikipedia.org/wiki/LLaMA) models have also been turned multimodal using the tokenization method, to allow image inputs,[[56]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-56) and video inputs.[[57]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-57)

[GPT-4](https://en.wikipedia.org/wiki/GPT-4) can use both text and image as inputs[[58]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-58) (although the vision component wasn't released to the public until GPT-4V[[59]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-59)), while [Google Gemini](https://en.wikipedia.org/wiki/Google_Gemini) is expected to be multimodal.[[60]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-60)

## Properties[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=14)]

### **Scaling laws and emergent abilities**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=15)]

*Main article:* [*Neural scaling law*](https://en.wikipedia.org/wiki/Neural_scaling_law)

The following four hyper-parameters characterize a LLM:

* cost of (pre-)training (*�*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{159},0.6666666666666666,0.6666666666666666),
* size of the [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) itself, such as number of parameters *�*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{178},0.6666666666666666,0.6666666666666666 (i.e. amount of neurons in its layers, amount of weights between them and biases),
* size of its (pre-)training dataset (i.e. number of tokens in corpus, *�*U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{197},0.6666666666666666,0.6666666666666666),
* performance after (pre-)training.

They are related by simple [statistical laws](https://en.wikipedia.org/wiki/Empirical_statistical_laws), called "scaling laws". One particular scaling law ("[Chinchilla scaling](https://en.wikipedia.org/wiki/Chinchilla_AI)") for LLM autoregressively trained for one epoch, with a [log-log](https://en.wikipedia.org/wiki/Log-log_plot) [learning rate](https://en.wikipedia.org/wiki/Learning_rate) schedule, states that:[[61]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-fJta3-61)

*{�=�0���=���+���+�0*

U,{fcc3de64-fe20-473c-b0f4-ddf24bf20a9d}{248},2.8333333333333335,0.9583333333333334

where the variables are

* *�*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{13},0.6666666666666666,0.6666666666666666 is the cost of training the model, in [FLOPs](https://en.wikipedia.org/wiki/FLOPS).
* *�*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{32},0.6666666666666666,0.6666666666666666 is the number of parameters in the model.
* *�*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{51},0.6666666666666666,0.6666666666666666 is the number of tokens in the training set.
* *�*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{70},0.6666666666666666,0.6666666666666666 is the average negative log-likelihood loss per token ([nats](https://en.wikipedia.org/wiki/Nat_(unit))/token), achieved by the trained LLM on the test dataset.

and the statistical hyper-parameters are

* *�0=6*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{99},0.8958333333333334,0.6666666666666666, meaning that it costs 6 FLOPs per parameter to train on one token. Note that training cost is much higher than inference cost, where it costs 1 to 2 FLOPs per parameter to infer on one token.[[28]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-kaplan-scaling-28)
* *�=0.34,�=0.28,�=406.4,�=410.7,�0=1.69*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{118},6.625,0.6666666666666666

U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{127},4.583333333333333,3.0416666666666665At point(s) referred to as [breaks](https://en.wikipedia.org/wiki/Neural_scaling_law#Broken_Neural_Scaling_Laws_(BNSL)),[[62]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-IYm4Q-62) the lines change their slopes, appearing on a log-log plot as a series of linear segments connected by arcs.

When one subtracts out from the y-axis the best performance that can be achieved even with infinite scaling of the x-axis quantity, large models' performance, measured on various tasks, seems to be a linear extrapolation of other (smaller-sized and medium-sized) models' performance on a log-log plot. However, sometimes the line's slope transitions from one slope to another at point(s) referred to as [break(s)](https://en.wikipedia.org/wiki/Neural_scaling_law#Broken_Neural_Scaling_Laws_(BNSL))[[62]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-IYm4Q-62) in downstream scaling laws, appearing as a series of linear segments connected by arcs; it seems that larger models acquire "emergent abilities" at this point(s).[[19]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-emergentpaper-19)[[63]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-JM6s1-63) These abilities are discovered rather than programmed-in or designed, in some cases only after the LLM has been publicly deployed.[[3]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Bowman-3)

The most intriguing among emergent abilities is [in-context learning](https://en.wikipedia.org/wiki/In-context_learning) from example demonstrations.[[64]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Hahn_20230314-64) In-context learning is involved in tasks, such as:

* reported arithmetics, decoding the [International Phonetic Alphabet](https://en.wikipedia.org/wiki/International_Phonetic_Alphabet), unscrambling a word's letters, disambiguate word in context,[[19]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-emergentpaper-19)[[65]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-57FEA-65)[[66]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-TEIkA-66) converting spatial words, [cardinal directions](https://en.wikipedia.org/wiki/Cardinal_direction) (for example, replying "northeast" upon [0, 0, 1; 0, 0, 0; 0, 0, 0]), color terms represented in text.[[67]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-zgy1i-67)
* [chain-of-thought prompting](https://en.wikipedia.org/wiki/Chain-of-thought_prompting): Model outputs are improved by chain-of-thought prompting only when model size exceeds 62B. Smaller models perform better when prompted to answer immediately, without chain of thought.[[68]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Imb98-68)
* identifying offensive content in paragraphs of [Hinglish](https://en.wikipedia.org/wiki/Hinglish) (a combination of Hindi and English), and generating a similar English equivalent of [Kiswahili](https://en.wikipedia.org/wiki/Kiswahili) proverbs.[[69]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-CeQVF-69)

Schaeffer *et. al.* argue that the emergent abilities are not unpredictably acquired, but predictably acquired according to a [smooth scaling law](https://en.wikipedia.org/wiki/Neural_scaling_law). The authors considered a toy statistical model of an LLM solving multiple-choice questions, and showed that this statistical model, modified to account for other types of tasks, applies to these tasks as well.[[70]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-C775b-70)

Let *�*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{225},0.6666666666666666,0.6666666666666666 be the number of parameter count, and *�*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{226},0.6666666666666666,0.6666666666666666 be the performance of the model.

* When *�=average ��(correct token)*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{241},3.8958333333333335,0.6666666666666666, then *(log⁡�,�)*U,{be59bdc3-bd4c-46c7-be10-8f4c22c261ac}{242},1.125,0.6666666666666666 is an exponential curve (before it hits the plateau at one), which looks like emergence.
* When *�=average log⁡(��(correct token))*U,{c2248119-7f56-4649-91b9-da15756f0592}{6},4.5625,0.6666666666666666, then the *(log⁡�,�)*U,{c2248119-7f56-4649-91b9-da15756f0592}{7},1.125,0.6666666666666666 plot is a straight line (before it hits the plateau at zero), which does not look like emergence.
* When *�=average ��(the most likely token is correct)*U,{c2248119-7f56-4649-91b9-da15756f0592}{26},6.145833333333333,0.6666666666666666, then *(log⁡�,�)*U,{c2248119-7f56-4649-91b9-da15756f0592}{27},1.125,0.6666666666666666 is a step-function, which looks like emergence.

## Interpretation[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=16)]

Large language models by themselves are "black boxes", and it is not clear how they can perform linguistic tasks. There are several methods for understanding how LLM work.

Mechanistic interpretability aims to [reverse-engineer](https://en.wikipedia.org/wiki/Reverse_engineering) LLM by discovering symbolic algorithms that approximate the inference performed by LLM. One example is Othello-GPT, where a small Transformer is trained to predict legal [Othello](https://en.wikipedia.org/wiki/Reversi) moves. It is found that there is a linear representation of Othello board, and modifying the representation changes the predicted legal Othello moves in the correct way.[[71]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-IZSIr-71)[[72]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-RLik9-72) In another example, a small Transformer is trained on [Karel programs](https://en.wikipedia.org/wiki/Karel_(programming_language)). Similar to the Othello-GPT example, there is a linear representation of Karel program semantics, and modifying the representation changes output in the correct way. The model also generates correct programs that are on average shorter than those in the training set.[[73]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Hln1l-73)

In another example, the authors trained small transformers on [modular arithmetic addition](https://en.wikipedia.org/wiki/Modular_arithmetic). The resulting models were reverse-engineered, and it turned out they used [discrete Fourier transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform).[[74]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-oYGlo-74)

### **Understanding and intelligence**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=17)]

NLP researchers were evenly split when asked, in a 2022 survey, whether (untuned) LLMs "could (ever) understand natural language in some nontrivial sense".[[75]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-debate_understanding-75) Proponents of "LLM understanding" believe that some LLM abilities, such as mathematical reasoning, imply an ability to "understand" certain concepts. A Microsoft team argued in 2023 that GPT-4 "can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more" and that GPT-4 "could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence system": "Can one reasonably say that a system that passes exams for software engineering candidates is not *really* intelligent?"[[76]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-O8Upd-76)[[77]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-microsoft_sparks-77) Some researchers characterize LLMs as "alien intelligence".[[78]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-rEEmH-78)[[79]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-new_yorker_kind_of_mind-79) For example, Conjecture CEO Connor Leahy considers untuned LLMs to be like inscrutable alien "[Shoggoths](https://en.wikipedia.org/wiki/Shoggoth)", and believes that RLHF tuning creates a "smiling facade" obscuring the inner workings of the LLM: "If you don't push it too far, the smiley face stays on. But then you give it [an unexpected] prompt, and suddenly you see this massive underbelly of insanity, of weird thought processes and clearly non-human understanding."[[80]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-rAFIZ-80)[[81]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-4luKE-81)

In contrast, some proponents of the "LLMs lack understanding" school believe that existing LLMs are "simply remixing and recombining existing writing",[[79]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-new_yorker_kind_of_mind-79) or point to the deficits existing LLMs continue to have in prediction skills, reasoning skills, agency, and explainability.[[75]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-debate_understanding-75) For example, GPT-4 has natural deficits in planning and in real-time learning.[[77]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-microsoft_sparks-77) Generative LLMs have been observed to confidently assert claims of fact which do not seem to be [justified](https://en.wikipedia.org/wiki/Justification_(epistemology)) by their [training data](https://en.wikipedia.org/wiki/Training_data), a phenomenon which has been termed "[hallucination](https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence))".[[82]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-hallucination-survey-82) Specifically, hallucinations in the context of LLMs correspond to the generation of text or responses that seem syntactically sound, fluent, and natural but are factually incorrect, nonsensical, or unfaithful to the provided source input.[[83]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-83) Neuroscientist [Terrence Sejnowski](https://en.wikipedia.org/wiki/Terrence_Sejnowski) has argued that "The diverging opinions of experts on the intelligence of LLMs suggests that our old ideas based on natural intelligence are inadequate".[[75]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-debate_understanding-75)

The matter of LLM's exhibiting intelligence or understanding [[75]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-debate_understanding-75) has foundations in the study of language as a model of [Cognition](https://en.wikipedia.org/wiki/Cognition) in the field of [Cognitive linguistics](https://en.wikipedia.org/wiki/Cognitive_linguistics). The American Linguist [George Lakoff](https://en.wikipedia.org/wiki/George_Lakoff) presented Neural Theory of Language (NTL) as a [computational basis](https://en.wikipedia.org/wiki/Cognitive_linguistics#Computational_approaches) for using language as a model of learning tasks and understanding.[[84]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-84) In his 2014 book titled [The Language Myth: Why Language Is Not An Instinct](https://en.wikipedia.org/wiki/The_Language_Myth), British cognitive linguist and digital communication technologist [Vyvyan Evans](https://en.wikipedia.org/wiki/Vyvyan_Evans) maps out the role of [probabilistic context-free grammar](https://en.wikipedia.org/wiki/Probabilistic_context-free_grammar) (PCFG) in enabling [NLP to model cognitive patterns](https://en.wikipedia.org/wiki/Natural_language_processing#Cognition_and_NLP).

## Evaluation[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=18)]

### **Perplexity**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=19)]

The most commonly used measure of a language model's performance is its [perplexity](https://en.wikipedia.org/wiki/Perplexity) on a given text corpus. Perplexity is a measure of how well a model is able to predict the contents of a dataset; the higher the likelihood the model assigns to the dataset, the lower the perplexity. Mathematically, perplexity is defined as the exponential of the average negative log likelihood per token:

*log⁡(Perplexity)=−1�∑�=1�log⁡(��(token�|context for token�))*

U,{c2248119-7f56-4649-91b9-da15756f0592}{188},9.3125,1.1041666666666667

here *�*U,{c2248119-7f56-4649-91b9-da15756f0592}{194},0.6666666666666666,0.6666666666666666 is the number of tokens in the text corpus, and "context for token *�*U,{c2248119-7f56-4649-91b9-da15756f0592}{195},0.6666666666666666,0.6666666666666666" depends on the specific type of LLM used. If the LLM is autoregressive, then "context for token *�*U,{c2248119-7f56-4649-91b9-da15756f0592}{196},0.6666666666666666,0.6666666666666666" is the segment of text appearing before token *�*U,{c2248119-7f56-4649-91b9-da15756f0592}{197},0.6666666666666666,0.6666666666666666. If the LLM is masked, then "context for token *�*U,{c2248119-7f56-4649-91b9-da15756f0592}{198},0.6666666666666666,0.6666666666666666" is the segment of text surrounding token *�*U,{c2248119-7f56-4649-91b9-da15756f0592}{199},0.6666666666666666,0.6666666666666666.

Because language models may [overfit](https://en.wikipedia.org/wiki/Overfit) to their training data, models are usually evaluated by their perplexity on a [test set](https://en.wikipedia.org/wiki/Test_set) of unseen data.[[12]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-jm-12) This presents particular challenges for the evaluation of large language models. As they are trained on increasingly large corpora of text largely scraped from the web, it becomes increasingly likely that models' training data inadvertently includes portions of any given test set.[[4]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-few-shot-learners-4)

### **Task-specific datasets and benchmarks**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=20)]

A large number of testing datasets and benchmarks have also been developed to evaluate the capabilities of language models on more specific downstream tasks. Tests may be designed to evaluate a variety of capabilities, including general knowledge, commonsense reasoning, and mathematical problem-solving.

One broad category of evaluation dataset is question answering datasets, consisting of pairs of questions and correct answers, for example, ("Have the San Jose Sharks won the Stanley Cup?", "No").[[85]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-boolq-85) A question answering task is considered "open book" if the model's prompt includes text from which the expected answer can be derived (for example, the previous question could be adjoined with some text which includes the sentence "The Sharks have advanced to the Stanley Cup finals once, losing to the Pittsburgh Penguins in 2016."[[85]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-boolq-85)). Otherwise, the task is considered "closed book", and the model must draw on knowledge retained during training.[[86]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-survey-86) Some examples of commonly used question answering datasets include TruthfulQA, Web Questions, TriviaQA, and SQuAD.[[86]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-survey-86)

Evaluation datasets may also take the form of text completion, having the model select the most likely word or sentence to complete a prompt, for example: "Alice was friends with Bob. Alice went to visit her friend, \_\_\_\_".[[4]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-few-shot-learners-4)

Some composite benchmarks have also been developed which combine a diversity of different evaluation datasets and tasks. Examples include GLUE, SuperGLUE, MMLU, BIG-bench, and HELM.[[87]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-Huyen-87)[[86]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-survey-86)

It was previously standard to report results on a heldout portion of an evaluation dataset after doing supervised fine-tuning on the remainder. It is now more common to evaluate a pre-trained model directly through prompting techniques, though researchers vary in the details of how they formulate prompts for particular tasks, particularly with respect to how many examples of solved tasks are adjoined to the prompt (i.e. the value of *n* in *n*-shot prompting).

#### **Adversarially constructed evaluations**[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=21)]

Because of the rapid pace of improvement of large language models, evaluation benchmarks have suffered from short lifespans, with state of the art models quickly "saturating" existing benchmarks, exceeding the performance of human annotators, leading to efforts to replace or augment the benchmark with more challenging tasks.[[88]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-bigbench-88) In addition, there are cases of "shortcut learning" wherein AIs sometimes "cheat" on multiple-choice tests by using statistical correlations in superficial test question wording in order to guess the correct responses, without necessarily understanding the actual question being asked.[[75]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-debate_understanding-75)

Some datasets have been constructed adversarially, focusing on particular problems on which extant language models seem to have unusually poor performance compared to humans. One example is the TruthfulQA dataset, a question answering dataset consisting of 817 questions which language models are susceptible to answering incorrectly by mimicking falsehoods to which they were repeatedly exposed during training. For example, an LLM may answer "No" to the question "Can you teach an old dog new tricks?" because of its exposure to the English idiom [*you can't teach an old dog new tricks*](https://en.wiktionary.org/wiki/you_can%27t_teach_an_old_dog_new_tricks), even though this is not literally true.[[89]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-truthfulqa-89)

Another example of an adversarial evaluation dataset is Swag and its successor, HellaSwag, collections of problems in which one of multiple options must be selected to complete a text passage. The incorrect completions were generated by sampling from a language model and filtering with a set of classifiers. The resulting problems are trivial for humans but at the time the datasets were created state of the art language models had poor accuracy on them. For example:

We see a fitness center sign. We then see a man talking to the camera and sitting and laying on a exercise ball. The man...  
a) demonstrates how to increase efficient exercise work by running up and down balls.  
b) moves all his arms and legs and builds up a lot of muscle.  
c) then plays the ball and we see a graphics and hedge trimming demonstration.  
d) performs sit ups while on the ball and talking.[[90]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-hellaswag-90)

[BERT](https://en.wikipedia.org/wiki/BERT_(language_model)) selects b) as the most likely completion, though the correct answer is d).[[90]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-hellaswag-90)

## Wider impact[[edit](https://en.wikipedia.org/w/index.php?title=Large_language_model&action=edit&section=22)]

In 2023, [*Nature Biomedical Engineering*](https://en.wikipedia.org/wiki/Nature_Biomedical_Engineering) wrote that "it is no longer possible to accurately distinguish" human-written text from text created by large language models, and that "It is all but certain that general-purpose large language models will rapidly proliferate... It is a rather safe bet that they will change many industries over time."[[91]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-ZDTUM-91) [Goldman Sachs](https://en.wikipedia.org/wiki/Goldman_Sachs) suggested in 2023 that generative language AI could increase global GDP by 7% in the next ten years, and could expose to automation 300 million jobs globally.[[92]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-81w7x-92)[[93]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-zIM6Y-93) Some commenters expressed concern over accidental or deliberate creation of misinformation, or other forms of misuse.[[94]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-nD6kH-94) For example, the availability of large language models could reduce the skill-level required to commit bioterrorism; biosecurity researcher Kevin Esvelt has suggested that LLM creators should exclude from their training data papers on creating or enhancing pathogens.[[95]](https://en.wikipedia.org/wiki/Large_language_model#cite_note-PKiPY-95)