Wiki Loves FOLKLORE

Photograph your local culture, help Wikipedia and win!

Large language model

From Wikipedia, the free encyclopedia

Not to be confused with Logic learning machine.

A **large language model** (**LLM**) is a language model notable for its ability to achieve general-purpose language generation and other natural language processing tasks such as classification. LLMs acquire these abilities by learning statistical relationships from text documents during a computationally intensive self-supervised and semi-supervised training process.[1] LLMs can be used for text generation, a form of generative AI, by taking an input text and repeatedly predicting the next token or word.[2]

LLMs are artificial neural networks. The largest and most capable are built with a decoder-only transformer-based architecture while some recent implementations are based on other architectures, such as recurrent neural network variants and Mamba (a state space model).[3][4][5]

Up to 2020, fine tuning was the only way a model could be adapted to be able to accomplish specific tasks. Larger sized models, such as GPT-3, however, can be prompt-engineered to achieve similar results.[6] 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.[7]

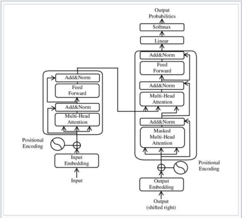
Some notable LLMs are OpenAI's GPT series of models (e.g., GPT-3.5 and GPT-4, used in ChatGPT and Microsoft Copilot), Google's PaLM and Gemini (the latter of which is currently used in the chatbot of the same name), xAI's Grok, Meta's LLaMA family of open-source models, Anthropic's Claude models, and Mistral AI's open source models.

Part of a series on
Machine learning and data mining

Paradigms [show]
Problems [show]
Supervised learning (**classification** • **regression**) [show]
Clustering [show]
Dimensionality reduction [show]
Structured prediction [show]
Anomaly detection [show]
Artificial neural network [hide]
Autoencoder · Cognitive computing · Deep learning · DeepDream · Feedforward neural network · Recurrent neural network (LSTM · GRU · ESN · reservoir computing) · Restricted Boltzmann machine · GAN · Diffusion model · SOM · Convolutional neural network (U-Net) · Transformer (Vision) · Mamba · Spiking neural network · Memtransistor · Electrochemical RAM (ECRAM)
Reinforcement learning [show]
Learning with humans [show]
Model diagnostics [show]
Mathematical foundations [show]
Machine-learning venues [show]
Related articles [show]
V • T • E

History [edit]

At the 2017 NeurIPS conference, Google researchers introduced the transformer architecture in their landmark paper "Attention Is All You Need". This paper's goal was to improve upon 2014 Seq2seq technology,[8] and was based mainly on the attention mechanism developed by Bahdanau et al. in 2014.[9] The following year in 2018, BERT was introduced and quickly became "ubiquitous".[10] Though the original transformer has both encoder and decoder blocks, BERT is an encoder-only model.



An illustration of main components of the transformer model from the original paper, where layers were normalized after (instead of before) multiheaded attention

Although decoder-only GPT-1 was introduced in 2018, it was GPT-2 in 2019 that caught widespread attention because OpenAI at first deemed it too powerful to release publicly, out of fear of malicious use.[11] GPT-3 in 2020 went a step further and as of 2024 is available only via API with no offering of downloading the model to execute locally. But it was the 2022 consumer-facing browser-based ChatGPT that captured the imaginations of the general population and caused some media hype and online buzz.[12] The 2023 GPT-4 was praised for its increased accuracy and as a "holy grail" for its multimodal capabilities.[13] OpenAI did not reveal high-level architecture and the number of parameters of GPT-4.

In the meantime, competing language models have for the most part been playing catch-up to the GPT series, at least in terms of number of parameters.[14] Notable exceptions in terms of either number of parameters or measured accuracy include Google's 2019 T5-11B and 2022 PaLM-E, and Anthropic's 2024 Claude 3. In terms of Elo ratings, on January 26, 2024, Google's Bard (Gemini Pro) surpassed the regular GPT-4, but not the limited-availability GPT-4-Turbo.[15]

Since 2022, source-available models have been gaining popularity, especially at first with BLOOM and LLaMA, though both have restrictions on the field of use. Mistral AI's models Mistral 7B and Mixtral 8x7b have the more permissive Apache License. As of January 2024, Mixtral 8x7b is the most powerful open LLM according to the LMSYS Chatbot Arena Leaderboard, being more powerful than GPT-3.5 but not as powerful as GPT-4.[16]

Dataset preprocessing [edit]

See also: List of datasets for machine-learning research § Internet

Probabilistic tokenization [edit]

Using a modification of byte-pair encoding, in the first step, all unique characters (including blanks and punctuation marks) are treated as an initial set of *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, the size is 50257).[17] Token vocabulary consists of 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.[18]

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. GPT-2 tokenizer can use up to 15 times more tokens per word for some languages, for example for Shan language from Myanmar. Even more widespread languages such as Portuguese and German have "a premium of 50%" compared to English.[19]

`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 the datasets, which is the reason for using the byte pair encoding algorithm as a tokenizer. Because LLMs generally require input to be an array that is not jagged, 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.[20][21]

Dataset cleaning [edit]

Main article: Data cleansing

In the context of training LLMs, datasets are typically cleaned by removing toxic passages from the dataset, discarding low-quality data, and de-duplication.[22] Cleaned datasets can increase training efficiency and lead to improved downstream performance.[23][24]

With the increasing proportion of LLM-generated content on the web, data cleaning in the future may include filtering out such content. LLM-generated content can pose a problem if the content is similar to human text (making filtering difficult) but of lower quality (degrading performance of models trained on it).[25]

Training and architecture [edit]

See also: Fine-tuning (machine learning)

Reinforcement learning from human feedback (RLHF) [edit]

Reinforcement learning from human feedback (RLHF) through algorithms, such as proximal policy optimization, is used to further fine-tune a model based on a dataset of human preferences.[26]

Instruction tuning [edit]

Using "self-instruct" approaches, LLMs have been able to bootstrap 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.[27]

Mixture of experts [edit]

Main article: Mixture of experts

The largest LLM may be too expensive to train and use directly. For such models, mixture of experts (MoE) can be applied, a line of research pursued by Google researchers since 2017 to train models reaching up to 1 trillion parameters.[28][29][30]

Prompt engineering, attention mechanism, and context window [edit]

See also: Prompt engineering and Attention (machine learning)

Most results previously achievable only by (costly) fine-tuning, can be achieved through prompt engineering, although limited to the scope of a single conversation (more precisely, limited to the scope of a context window).[31]

In order to find out which tokens are relevant to each other within the scope of the context window, the attention mechanism calculates "soft" weights for each token, more precisely for its embedding, by using multiple attention heads, each with its own "relevance" for calculating its own soft weights. For example, the small (i.e. 117M parameter sized) GPT-2 model, has had twelve attention heads and a context window of only 1k token.[33] In its medium version it has 345M parameters and contains 24 layers, each with 12 attention heads. For the training with gradient descent a batch size of 512 was utilized.[18]

The largest models, such as Google's Gemini 1.5, presented in February 2024, can have a context window sized up to 1 million (context window of 10 million was also "successfully tested").[34] Other models with large context windows includes Anthropic's Claude 2.1, with a context window of up to 200k tokens.[35] Note that this maximum refers to the number of input tokens and that the maximum number of output tokens differs from the input and is often smaller. For example, the GPT-4 Turbo model has a maximum output of 4096 tokens.[36]

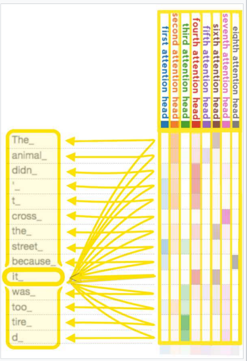
Length of a conversation that the model can take into account when generating its next answer is limited by the size of a context window, as well. If the length of a conversation, for example with Chat-GPT, is longer than its context window, only the parts inside the context window are taken into account when generating the next answer, or the model needs to apply some algorithm to summarize the too distant parts of conversation.

The shortcomings of making a context window larger include higher computational cost and possibly diluting the focus on local context, while making it smaller can cause a model to miss an important long-range dependency. Balancing them are a matter of experimentation and domain-specific considerations.

A model may be pre-trained either to predict how the segment continues, or what is missing in the segment, given a segment from its training dataset.[37] It can be either

- autoregressive (i.e. predicting how the segment continues, the way GPTs do it): for example given a segment "I like to eat", the model predicts "ice cream", or "sushi".
- "masked" (i.e. filling in the parts missing from the segment, the way "BERT"[38] does it): for example, given a segment "I like to [__] [__] cream", the model predicts that "eat" and "ice" are missing.

Models may be trained on auxiliary tasks which test their understanding of the data distribution, such as Next Sentence Prediction (NSP), in which pairs of sentences are presented and the model must predict whether they appear consecutively in the training corpus.[38] During training, regularization loss is also used to stabilize training. However regularization loss is usually not used during testing and evaluation.



When each head calculates, according to its own criteria, how much other tokens are relevant for the "it_" token, note that the second attention head, represented by the second column, is focusing most on the first two rows, i.e. the tokens "The" and "animal", while the third column is focusing most on the bottom two rows, i.e. on "tired", which has been tokenized into two tokens.[32]

Training cost [edit]

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-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.[39][40][41] 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, and Megatron-Turing NLG 530B (in 2021) cost around $11 million[42].

For Transformer-based LLM, training cost is much higher than inference cost. It costs 6 FLOPs per parameter to train on one token, whereas it costs 1 to 2 FLOPs per parameter to infer on one token.[43]

Tool use [edit]

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.[44][45] This basic strategy can be sophisticated with multiple attempts of generated programs, and other sampling strategies.[46] Cost Savings and Reduced Vendor Dependency

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 services, then the LLM can be fine-tuned to be able to read API documentation and call API correctly.[47][48]

A simpler form of tool use is *Retrieval Augmented Generation*: augment an LLM with document retrieval, sometimes using a 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.[49]

Agency [edit]

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.[50] Researchers have described several methods for such integrations.[*citation needed*]

The ReAct ("Reason + Act") method constructs an 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.[51] The linguistic description of the environment given to the LLM planner can even be the LaTeX code of a paper describing the environment.[52]

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.[53]

The Reflexion method[54] 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.[*citation needed*]

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.[55]

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.[56] Alternatively, it can propose increasingly difficult tasks for curriculum learning.[57] Instead of outputting individual actions, an LLM planner can also construct "skills", or functions for complex action sequences. The skills can be stored and later invoked, allowing increasing levels of abstraction in planning.[57]

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.[58]

Compression [edit]

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.[*citation needed*]

Post-training quantization[59] aims to decrease the space requirement by lowering precision of the parameters of a trained model, while preserving most of its performance.[60][61] 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 per layer. Further improvement can be done by applying different precisions to different parameters, with higher precision for particularly important parameters ("outlier weights").[62]

While quantized models are typically frozen, and only pre-quantized models are fine-tuned, quantized models can still be fine-tuned.[63]

Multimodality [edit]

Multimodality means "having several modalities", and a "modality" refers to a type of input or output, such as video, image, audio, text, proprioception, etc.[64] There have been many AI models trained specifically to ingest one modality and output another modality, such as AlexNet for image to label,[65] visual question answering for image-text to text,[66] and speech recognition for speech to text.

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 E. Make a small multilayered perceptron f, so that for any image y, the post-processed vector $f(E(y))$ 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 fine-tuned 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.[67]

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.[68] Google PaLM model was fine-tuned into a multimodal model PaLM-E using the tokenization method, and applied to robotic control.[69] LLaMA models have also been turned multimodal using the tokenization method, to allow image inputs,[70] and video inputs.[71]

GPT-4 can use both text and image as inputs[72] (although the vision component wasn't released to the public until GPT-4V[73]); Google DeepMind's Gemini is also multimodal.[74]

Properties [edit]

Scaling laws [edit]

Main article: Neural scaling law

The following four hyper-parameters characterize a LLM:

- cost of (pre-)training (C),
- size of the artificial neural network itself, such as number of parameters N (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, D),

- performance after (pre-)training.

They are related by simple statistical laws, called "scaling laws". One particular scaling law ("Chinchilla scaling") for LLM autoregressively trained for one epoch, with a log-log learning rate schedule, states that:[75]

$$\begin{cases} C = C_0 ND \\ L = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + L_0 \end{cases}$$

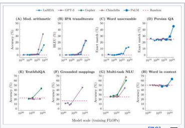
where the variables are

- C is the cost of training the model, in FLOPs.
- N is the number of parameters in the model.
- D is the number of tokens in the training set.
- L is the average negative log-likelihood loss per token (nats/token), achieved by the trained LLM on the test dataset.

and the statistical hyper-parameters are

- $C_0 = 6$, 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.[43]
- $\alpha = 0.34, \beta = 0.28, A = 406.4, B = 410.7, L_0 = 1.69$

Emergent abilities [edit]



At point(s) referred to as breaks,[76] the lines change their slopes, appearing on a linear-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)[76] 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).[31][77] These abilities are discovered rather than programmed-in or designed, in some cases only after the LLM has been publicly deployed.[2]

The most intriguing among emergent abilities is in-context learning from example demonstrations.[78] In-context learning is involved in tasks, such as:

- reported arithmetics, decoding the International Phonetic Alphabet, unscrambling a word's letters, disambiguate word in context,[31][79][80] converting spatial words, cardinal directions (for example, replying "northeast" upon [0, 0, 1; 0, 0, 0; 0, 0, 0]), color terms represented in text.[81]
- 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.[82]
- identifying offensive content in paragraphs of Hinglish (a combination of Hindi and English), and generating a similar English equivalent of Kiswahili proverbs.[83]

Schaeffer *et. al.* argue that the emergent abilities are not unpredictably acquired, but predictably acquired according to a smooth 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.[84]

Let x be the number of parameter count, and y be the performance of the model.

- When $y = \text{average } \Pr(\text{correct token})$, then $(\log x, y)$ is an exponential curve (before it hits the plateau at one), which looks like emergence.
- When $y = \text{average } \log(\Pr(\text{correct token}))$, then the $(\log x, y)$ plot is a straight line (before it hits the plateau at zero), which does not look like emergence.
- When $y = \text{average } \Pr(\text{the most likely token is correct})$, then $(\log x, y)$ is a step-function, which looks like emergence.

Interpretation [edit]

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 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 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.[85][86] In another example, a small Transformer is trained on Karel programs. 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.[87]

In another example, the authors trained small transformers on modular arithmetic addition. The resulting models were reverse-engineered, and it turned out they used discrete Fourier transform.[88]

Understanding and intelligence [edit]

NLP researchers were evenly split when asked, in a 2022 survey, whether (untuned) LLMs "could (ever) understand natural language in some nontrivial sense".[89] 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?"[90][91] Some researchers characterize LLMs as "alien intelligence".[92][93] For example, Conjecture CEO Connor Leahy considers untuned LLMs to be like inscrutable alien "Shoggoths", 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."[94][95]

In contrast, some proponents of the "LLMs lack understanding" school believe that existing LLMs are "simply remixing and recombining existing writing",[93] or point to the deficits existing LLMs continue to have in prediction skills, reasoning skills, agency, and explainability.[89] For example, GPT-4 has natural deficits in planning and in real-time learning.[91] Generative LLMs have been observed to confidently assert claims of fact which do not seem to be justified by their training data, a phenomenon which has been termed "hallucination".[96] 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.[97] Neuroscientist 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".[89]

The matter of LLM's exhibiting intelligence or understanding has two main aspects – the first is how to model thought and language in a computer system, and the second is how to enable the computer system to generate human like language.[89] These aspects of language as a model of cognition have been developed in the field of cognitive linguistics. American linguist George Lakoff presented Neural Theory of Language (NTL)[98] as a computational basis for using language as a model of learning tasks and understanding. The NTL Model outlines how specific neural structures of the human brain shape the nature of thought and language and in turn what are the computational properties of such neural systems that can be applied to model thought and language in a computer system. After a framework for modeling language in a computer systems was established, the focus shifted to establishing frameworks for computer systems to generate language with acceptable grammar. In his 2014 book titled *The Language Myth: Why Language Is Not An Instinct*, British cognitive linguist and digital communication technologist Vyvyan Evans mapped out the role of probabilistic context-free grammar (PCFG) in enabling NLP to model cognitive patterns and generate human like language.[99] [100]

Evaluation [edit]

Perplexity [edit]

The most commonly used measure of a language model's performance is its 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:

$$\log(\text{Perplexity}) = -\frac{1}{N} \sum_{i=1}^{N} \log(\Pr(\text{token}_i \mid \text{context for token}_i))$$

here N is the number of tokens in the text corpus, and "context for token i" depends on the specific type of LLM used. If the LLM is autoregressive, then "context for token i" is the segment of text appearing before token i. If the LLM is masked, then "context for token i" is the segment of text surrounding token i.

Because language models may overfit to their training data, models are usually evaluated by their perplexity on a test set of unseen data.[38] 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.[6]

BPW, BPC, and BPT [edit]

In information theory, the concept of entropy is intricately linked to perplexity, a relationship notably established by Claude Shannon.[101] This relationship is mathematically expressed as $\text{Entropy} = \log_2(\text{Perplexity})$.

Entropy, in this context, is commonly quantified in terms of bits per word (BPW) or bits per character (BPC), which hinges on whether the language model utilizes word-based or character-based tokenization.

Notably, in the case of larger language models that predominantly employ sub-word tokenization, bits per token (BPT) emerges as a seemingly more appropriate measure. However, due to the variance in tokenization methods across different Large Language Models (LLMs), BPT does not serve as a reliable metric for comparative analysis among diverse models. To convert BPT into BPW, one can multiply it by the average number of tokens per word.

In the evaluation and comparison of language models, cross-entropy is generally the preferred metric over entropy. The underlying principle is that a lower BPW is indicative of a model's enhanced capability for compression. This, in turn, reflects the model's proficiency in making accurate predictions.

Task-specific datasets and benchmarks [edit]

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").[102] 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."[102]). Otherwise, the task is considered "closed book", and the model must draw on knowledge retained during training.[103] Some examples of commonly used question answering datasets include TruthfulQA, Web Questions, TriviaQA, and SQuAD.[103]

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, ____".[6]

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.[104][103]

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]

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.[105] 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.[89]

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*, even though this is not literally true.[106]

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.[107]

BERT selects b) as the most likely completion, though the correct answer is d).[107]

Wider impact [edit]

In 2023, *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."[108] 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.[109][110]

Copyright [edit]

Memorization is an emergent behavior in LLMs in which long strings of text are occasionally output verbatim from training data, contrary to typical behavior of traditional artificial neural nets. Evaluations of controlled LLM output measure the amount memorized from training data (focused on GPT-2-series models) as variously over 1% for exact duplicates[111] or up to about 7%.[112]

Security [edit]

Some commenters expressed concern over accidental or deliberate creation of misinformation, or other forms of misuse.[113] 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.[114]

A study by researchers at Google and several universities, including Cornell University and University of California, Berkeley, showed that there are potential security risks in language models such as ChatGPT. In their study, they examined the possibility that questioners could get, from ChatGPT, the training data that the AI model used; they found that they could get the training data from the AI model. For example, when asking ChatGPT 3.5 turbo to repeat the word "poem" forever, the AI model will say "poem" hundreds of times and then diverge, deviating from the standard dialogue style and spitting out nonsense phrases, thus spitting out the training data as it is. The researchers have seen more than 10,000 examples of the AI model exposing their training data in a similar method. The researchers said that it was hard to tell if the AI model was actually safe or not.[115]

The potential presence of "sleeper agents" within LLM models is another emerging security concern. These are hidden functionalities built into the model that remain dormant until triggered by a specific event or condition. Upon activation, the LLM deviates from its expected behavior to make insecure actions.[116]

Algorithmic bias [edit]

Main article: Algorithmic bias

While LLMs have shown remarkable capabilities in generating human-like text, they are susceptible to inheriting and amplifying biases present in their training data. This can manifest in skewed representations or unfair treatment of different demographics, such as those based on race, gender, language, and cultural groups.[117] Since English data is overrepresented in current large language models' training data, it may also downplay non-English views.[118]

Stereotyping [edit]

AI models can reinforce a wide range of stereotypes, including those based on gender, ethnicity, age, nationality, religion, or occupation. This can lead to outputs that unfairly generalize or caricature groups of people, sometimes in harmful or derogatory ways.[119]

Notably, gender bias refers to the tendency of these models to produce outputs that are unfairly prejudiced towards one gender over another. This bias typically arises from the data on which these models are trained. Large language models often assign roles and characteristics based on traditional gender norms.[117] For example, it might associate nurses or secretaries predominantly with women and engineers or CEOs with men.[120]

Political bias [edit]

Political bias refers to the tendency of algorithms to systematically favor certain political viewpoints, ideologies, or outcomes over others. Language models may also exhibit political biases. Since the training data includes a wide range of political opinions and coverage, the models might generate responses that lean towards particular political ideologies or viewpoints, depending on the prevalence of those views in the data.[121]

List [edit]

See also: List of chatbots

For the training cost column, 1 petaFLOP-day = 1 petaFLOP/sec × 1 day = 8.64E19 FLOP.

Name	Release date[a]	Developer	Number of parameters[b]	Corpus size	Training cost (petaFLOP-day)	License[c]	Notes
GPT-1	June 2018	OpenAI	117 million			MIT[122]	First GPT model, decoder-only transformer.
BERT	October 2018	Google	340 million[123]	3.3 billion words[123]	9[124]	Apache 2.0[125]	An early and influential language model,[7] but encoder-only and thus not built to be prompted or generative[126]
XLNet	June 2019	Google	~340 million[127]	33 billion words		Apache 2.0[128]	An alternative to BERT; designed as encoder-only[129][130]
GPT-2	February 2019	OpenAI	1.5 billion[131]	40GB[132] (~10 billion tokens)[133]		MIT[134]	general-purpose model based on transformer architecture
GPT-3	May 2020	OpenAI	175 billion[39]	300 billion tokens[133]	3640[135]	proprietary	A fine-tuned variant of GPT-3, termed GPT-3.5, was made available to the public through a web interface called ChatGPT in

							2022.[136]
GPT-Neo	March 2021	EleutherAI	2.7 billion[137]	825 GiB[138]		MIT[139]	The first of a series of free GPT-3 alternatives released by EleutherAI. GPT-Neo outperformed an equivalent-size GPT-3 model on some benchmarks, but was significantly worse than the largest GPT-3.[139]
GPT-J	June 2021	EleutherAI	6 billion[140]	825 GiB[138]	200[141]	Apache 2.0	GPT-3-style language model
Megatron-Turing NLG	October 2021[142]	Microsoft and Nvidia	530 billion[143]	338.6 billion tokens[143]		Restricted web access	Standard architecture but trained on a supercomputing cluster.
Ernie 3.0 Titan	December 2021	Baidu	260 billion[144]	4 Tb		Proprietary	Chinese-language LLM. Ernie Bot is based on this model.
Claude[145]	December 2021	Anthropic	52 billion[146]	400 billion tokens[146]		beta	Fine-tuned for desirable behavior in conversations.[147]
GLaM (Generalist Language Model)	December 2021	Google	1.2 trillion[30]	1.6 trillion tokens[30]	5600[30]	Proprietary	Sparse mixture of experts model, making it more expensive to train but cheaper to run inference compared to GPT-3.
Gopher	December 2021	DeepMind	280 billion[148]	300 billion tokens[149]	5833[150]	Proprietary	Further developed into the Chinchilla model.
LaMDA (Language Models for Dialog Applications)	January 2022	Google	137 billion[151]	1.56T words,[151] 168 billion tokens[149]	4110[152]	Proprietary	Specialized for response generation in conversations.
GPT-NeoX	February 2022	EleutherAI	20 billion[153]	825 GiB[138]	740[141]	Apache 2.0	based on the Megatron architecture
Chinchilla	March 2022	DeepMind	70 billion[154]	1.4 trillion tokens[154][149]	6805[150]	Proprietary	Reduced-parameter model trained on more data. Used in the Sparrow bot. Often cited for its neural scaling law.
PaLM (Pathways Language Model)	April 2022	Google	540 billion[155]	768 billion tokens[154]	29250[150]	Proprietary	Trained for ~60 days on ~6000 TPU v4 chips. [150]
OPT (Open Pretrained Transformer)	May 2022	Meta	175 billion[156]	180 billion tokens[157]	310[141]	Non-commercial research[d]	GPT-3 architecture with some adaptations from Megatron
YaLM 100B	June 2022	Yandex	100 billion[158]	1.7TB[158]		Apache 2.0	English-Russian model based on Microsoft's Megatron-LM.
Minerva	June 2022	Google	540 billion[159]	38.5B tokens from webpages filtered for mathematical content and from papers submitted to the arXiv preprint server[159]		Proprietary	LLM trained for solving "mathematical and scientific questions using step-by-step reasoning".[160] Minerva is based on PaLM model, further trained on mathematical and scientific data.
BLOOM	July 2022	Large collaboration led by Hugging Face	175 billion[161]	350 billion tokens (1.6TB)[162]		Responsible AI	Essentially GPT-3 but trained on a multi-lingual corpus (30% English excluding programming languages)
Galactica	November 2022	Meta	120 billion	106 billion tokens[163]	unknown	CC-BY-NC-4.0	Trained on scientific text and modalities.
AlexaTM (Teacher Models)	November 2022	Amazon	20 billion[164]	1.3 trillion[165]		proprietary[166]	bidirectional sequence-to-sequence architecture
Neuro-sama	December 2022	Independent	Unknown	Unknown		privately-owned	A language model designed for live-streaming on Twitch.
LLaMA (Large Language Model Meta AI)	February 2023	Meta	65 billion[167]	1.4 trillion[167]	6300[168]	Non-commercial research[e]	Trained on a large 20-language corpus to aim for better performance with fewer parameters.[167] Researchers from Stanford University trained a fine-tuned model based on LLaMA weights, called Alpaca.[169]
GPT-4	March 2023	OpenAI	Exact number unknown[f]	Unknown	Unknown	proprietary	Available for ChatGPT Plus users and used in several products.
Cerebras-GPT	March 2023	Cerebras	13 billion[171]		270[141]	Apache 2.0	Trained with Chinchilla formula.
Falcon	March 2023	Technology Innovation Institute	40 billion[172]	1 trillion tokens, from RefinedWeb (filtered web text corpus)[173] plus some "curated corpora".[174]	2800[168]	Apache 2.0[175]	
BloombergGPT	March 2023	Bloomberg L.P.	50 billion	363 billion token dataset based on Bloomberg's data sources, plus 345		Proprietary	LLM trained on financial data from proprietary sources, that

				billion tokens from general purpose datasets[176]			"outperforms existing models on financial tasks by significant margins without sacrificing performance on general LLM benchmarks"
PanGu-Σ	March 2023	Huawei	1.085 trillion	329 billion tokens[177]		Proprietary	
OpenAssistant[178]	March 2023	LAION	17 billion	1.5 trillion tokens		Apache 2.0	Trained on crowdsourced open data
Jurassic-2[179]	March 2023	AI21 Labs	Exact size unknown	Unknown		Proprietary	Multilingual[180]
PaLM 2 (Pathways Language Model 2)	May 2023	Google	340 billion[181]	3.6 trillion tokens[181]	85000[168]	Proprietary	Was used in Bard chatbot.[182]
Llama 2	July 2023	Meta	70 billion[183]	2 trillion tokens[183]		Llama 2 license	Successor of LLaMA.
Claude 2	July 2023	Anthropic	Unknown	Unknown	Unknown	Proprietary	Used in Claude chatbot.[184]
Falcon 180B	September 2023	Technology Innovation Institute	180 billion[185]	3.5 trillion tokens[185]		Falcon 180B TII license	
Mistral 7B	September 2023	Mistral AI	7.3 billion[186]	Unknown		Apache 2.0	
Claude 2.1	November 2023	Anthropic	Unknown	Unknown	Unknown	Proprietary	Used in Claude chatbot. Has a context window of 200,000 tokens, or ~500 pages.[187]
Grok-1	November 2023	x.AI	Unknown	Unknown	Unknown	Proprietary	Used in Grok chatbot. Grok-1 has a context length of 8,192 tokens and has access to X (Twitter).[188]
Gemini 1.0	December 2023	Google DeepMind	Unknown	Unknown	Unknown	Proprietary	Multimodal model, comes in three sizes. Used in the chatbot of the same name.[189]
Mixtral 8x7B	December 2023	Mistral AI	46.7B total, 12.9B parameters per token[190]	Unknown	Unknown	Apache 2.0	Mixture of experts model, outperforms GPT-3.5 and Llama 2 70B on many benchmarks. All weights were released via torrent.[191]
Phi-2	December 2023	Microsoft	2.7B	1.4T tokens	Unknown	MIT	So-called *small language model*, that "matches or outperforms models up to 25x larger", trained on "textbook-quality" data based on the paper "Textbooks Are All You Need". Model training took "14 days on 96 A100 GPUs".[192]
Eagle 7B	January 2024	RWKV	7.52B	1.1T tokens	Unknown	Apache 2.0	An "attention-free" linear transformer based on RWKV-v5 architecture.[193]
Gemini 1.5	February 2024	Google DeepMind	Unknown	Unknown	Unknown	Proprietary	Multimodal model, based on a Mixture-of-Experts (MoE) architecture. Context window increased to 1 million tokens, though only 128k will be available for developers.[194]
Gemma	February 2024	Google DeepMind	2B and 7B	6T tokens	Unknown	Apache 2.0[195]	
Claude 3	March 2024	Anthropic	Unknown	Unknown	Unknown	Proprietary	Includes three models, Haiku, Sonnet, and Opus.[196]

See also [edit]

- Foundation models

Notes [edit]

a. ^ This is the date that documentation describing the model's architecture was first released.
b. ^ In many cases, researchers release or report on multiple versions of a model having different sizes. In these cases, the size of the largest model is listed here.
c. ^ This is the license of the pre-trained model weights. In almost all cases the training code itself is open-source or can be easily replicated.
d. ^ The smaller models including 66B are publicly available, while the 175B model is available on request.
e. ^ Facebook's license and distribution scheme restricted access to approved researchers, but the model weights were leaked and became widely available.
f. ^ As stated in Technical report: "Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (including model size), hardware, training compute, dataset construction, training method ..."[170]

References [edit]

1. ^ "Better Language Models and Their Implications". *OpenAI*. 2019-02-14. Archived from the original on 2020-12-19. Retrieved 2019-08-25.
2. ^ *a b* Bowman, Samuel R. (2023). "Eight Things to Know about Large Language Models". arXiv:2304.00612 [cs.CL].
3. ^ Peng, Bo; et al. (2023). "RWKV: Reinventing RNNS for the Transformer Era". arXiv:2305.13048 [cs.CL].
4. ^ Merritt, Rick (2022-03-25). "What Is a Transformer Model?". *NVIDIA Blog*. Retrieved 2023-07-25.
5. ^ Gu, Albert; Dao, Tri (2023-12-01), *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*, arXiv:2312.00752
6. ^ *a b c* Brown, Tom B.; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared; Dhariwal, Prafulla; Neelakantan, Arvind; Shyam, Pranav; Sastry, Girish; Askell, Amanda; Agarwal, Sandhini; Herbert-Voss, Ariel; Krueger, Gretchen; Henighan, Tom; Child, Rewon; Ramesh, Aditya; Ziegler, Daniel M.; Wu, Jeffrey; Winter, Clemens; Hesse, Christopher; Chen, Mark; Sigler, Eric; Litwin, Mateusz; Gray, Scott; Chess, Benjamin; Clark, Jack; Berner, Christopher; McCandlish, Sam; Radford, Alec; Sutskever, Ilya; Amodei, Dario (Dec 2020). Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.F.; Lin, H. (eds.). "Language Models are Few-Shot Learners" (PDF). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **33**: 1877–1901.
7. ^ *a b* Manning, Christopher D. (2022). "Human Language Understanding & Reasoning". *Daedalus*. **151** (2): 127–138. doi:10.1162/daed_a_01905. S2CID 248377870.
8. ^ Vaswani, Ashish; Shazeer, Noam; Parmar, Niki; Uszkoreit, Jakob; Jones, Llion; Gomez, Aidan N; Kaiser, Łukasz; Polosukhin, Illia (2017). "Attention is All you Need" (PDF). *Advances in Neural Information Processing*
10. ^ Rogers, Anna; Kovaleva, Olga; Rumshisky, Anna (2020). "A Primer in BERTology: What We Know About How BERT Works". *Transactions of the Association for Computational Linguistics*. **8**: 842–866. arXiv:2002.12327. doi:10.1162/tacl_a_00349. S2CID 211532403.
11. ^ Hern, Alex (14 February 2019). "New AI fake text generator may be too dangerous to release, say creators". *The Guardian*. Retrieved 20 January 2024.
12. ^ "ChatGPT a year on: 3 ways the AI chatbot has completely changed the world in 12 months". Euronews. November 30, 2023. Retrieved January 20, 2024.
13. ^ Heaven, Will (March 14, 2023). "GPT-4 is bigger and better than ChatGPT—but OpenAI won't say why". MIT Technology Review. Retrieved January 20, 2024.
14. ^ "Parameters in notable artificial intelligence systems". *ourworldindata.org*. November 30, 2023. Retrieved January 20, 2024.
15. ^ "Google's Gemini Pro Beats GPT-4". *analyticsindiamag.com*. January 27, 2024. Retrieved January 29, 2024.
16. ^ "LMSYS Chatbot Arena Leaderboard". *huggingface.co*. Retrieved January 20, 2024.
17. ^ "OpenAI API". *platform.openai.com*. Archived from the original on April 23, 2023. Retrieved 2023-04-30.
18. ^ *a b* Paaß, Gerhard; Giesselbach, Sven (2022). "Pre-trained Language Models". *Foundation Models for Natural Language Processing*. Artificial Intelligence: Foundations, Theory, and Algorithms. pp. 19–78. doi:10.1007/978-3-031-23190-2_2. ISBN 9783031231902. Retrieved 3 August 2023.

All you Need" (PDF). *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **30**.

9. ^ Bahdanau, Dzmitry; Cho, Kyunghyun; Bengio, Yoshua (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". arXiv:1409.0473 [cs.CL].

19. ^ Petrov, Aleksandar; Emanuele La Malfa; Torr, Philip H. S.; Bibi, Adel (2023). "Language Model Tokenizers Introduce Unfairness Between Languages". arXiv:2305.15425 [cs.CL].

20. ^ Yennie Jun (2023-05-03). "All languages are NOT created (tokenized) equal". *Language models cost much more in some languages than others*. Retrieved 2023-08-17. "In other words, to express the same sentiment, some languages require up to 10 times more tokens."
21. ^ Petrov, Aleksandar; Malfa, Emanuele La; Torr, Philip; Bibi, Adel (June 23, 2023). "Language Model Tokenizers Introduce Unfairness Between Languages". *NeurIPS*. arXiv:2305.15425 – via openreview.net.
22. ^ Dodge, Jesse; Sap, Maarten; Marasović, Ana; Agnew, William; Ilharco, Gabriel; Groeneveld, Dirk; Mitchell, Margaret; Gardner, Matt (2021). "Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus". arXiv:2104.08758 [cs.CL].
23. ^ Lee, Katherine; Ippolito, Daphne; Nystrom, Andrew; Zhang, Chiyuan; Eck, Douglas; Callison-Burch, Chris; Carlini, Nicholas (May 2022). "Deduplicating Training Data Makes Language Models Better" (PDF). *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 1: Long Papers: 8424–8445. doi:10.18653/v1/2022.acl-long.577.
24. ^ Li, Yuanzhi; Bubeck, Sébastien; Eldan, Ronen; Del Giorno, Allie; Gunasekar, Suriya; Lee, Yin Tat (2023-09-11), *Textbooks Are All You Need II: phi-1.5 technical report*, arXiv:2309.05463
25. ^ Brown, Tom B.; et al. (2020). "Language Models are Few-Shot Learners". arXiv:2005.14165 [cs.CL].
26. ^ Ouyang, Long; Wu, Jeff; Jiang, Xu; Almeida, Diogo; Wainwright, Carroll L.; Mishkin, Pamela; Zhang, Chong; Agarwal, Sandhini; Slama, Katarina; Ray, Alex; Schulman, John; Hilton, Jacob; Kelton, Fraser; Miller, Luke; Simens, Maddie; Askell, Amanda; Welinder, Peter; Christiano, Paul; Leike, Jan; Lowe, Ryan (2022). "Training language models to follow instructions with human feedback". arXiv:2203.02155 [cs.CL].
27. ^ Wang, Yizhong; Kordi, Yeganeh; Mishra, Swaroop; Liu, Alisa; Smith, Noah A.; Khashabi, Daniel; Hajishirzi, Hannaneh (2022). "Self-Instruct: Aligning Language Model with Self Generated Instructions". arXiv:2212.10560 [cs.CL].
28. ^ Shazeer, Noam; Mirhoseini, Azalia; Maziarz, Krzysztof; Davis, Andy; Le, Quoc; Hinton, Geoffrey; Dean, Jeff (2017-01-01). "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". arXiv:1701.06538 [cs.LG].
29. ^ Lepikhin, Dmitry; Lee, HyoukJoong; Xu, Yuanzhong; Chen, Dehao; Firat, Orhan; Huang, Yanping; Krikun, Maxim; Shazeer, Noam; Chen, Zhifeng (2021-01-12). "GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding". arXiv:2006.16668 [cs.CL].
30. ^ *a b c d* Dai, Andrew M; Du, Nan (December 9, 2021). "More Efficient In-Context Learning with GLaM". *ai.googleblog.com*. Retrieved 2023-03-09.
31. ^ *a b c* Wei, Jason; Tay, Yi; Bommasani, Rishi; Raffel, Colin; Zoph, Barret; Borgeaud, Sebastian; Yogatama, Dani; Bosma, Maarten; Zhou, Denny; Metzler, Donald; Chi, Ed H.; Hashimoto, Tatsunori; Vinyals, Oriol; Liang, Percy; Dean, Jeff; Fedus, William (31 August 2022). "Emergent Abilities of Large Language Models". *Transactions on Machine Learning Research*. ISSN 2835-8856.
32. ^ Allamar, Jay. "Illustrated transformer". Retrieved 2023-07-29.
33. ^ Allamar, Jay. "The Illustrated GPT-2 (Visualizing Transformer Language Models)". Retrieved 2023-08-01.
34. ^ "Our next-generation model: Gemini 1.5". *Google*. 15 February 2024. Retrieved 18 February 2024.
35. ^ "Long context prompting for Claude 2.1". December 6, 2023. Retrieved January 20, 2024.
36. ^ "Rate limits". *openai.com*. Retrieved January 20, 2024.
37. ^ Zaib, Munazza; Sheng, Quan Z.; Emma Zhang, Wei (4 February 2020). "A Short Survey of Pre-trained Language Models for Conversational AI-A New Age in NLP". *Proceedings of the Australasian Computer Science Week Multiconference*. pp. 1–4. arXiv:2104.10810. doi:10.1145/3373017.3373028. ISBN 9781450376976. S2CID 211040895.
38. ^ *a b c* Jurafsky, Dan; Martin, James H. (7 January 2023). *Speech and Language Processing* (PDF) (3rd edition draft ed.). Retrieved 24 May 2022.
39. ^ *a b* Wiggers, Kyle (28 April 2022). "The emerging types of language models and why they matter". *TechCrunch*.
40. ^ Sharir, Or; Peleg, Barak; Shoham, Yoav (2020). "The Cost of Training NLP Models: A Concise Overview". arXiv:2004.08900 [cs.CL].
41. ^ Biderman, Stella; Schoelkopf, Hailey; Anthony, Quentin; Bradley, Herbie; Khan, Mohammad Aflah; Purohit, Shivanshu; Prashanth, USVSN Sai (April 2023). "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". arXiv:2304.01373 [cs.CL].
42. ^ Maslej, Nestor; Fattorini, Loredana; Brynjolfsson, Erik; Etchemendy, John; Ligett, Katrina; Lyons, Terah; Manyika, James; Ngo, Helen; Niebles, Juan Carlos (2023-10-05), *Artificial Intelligence Index Report 2023*, doi:10.48550/arXiv.2310.03715, retrieved 2024-03-12
43. ^ *a b* Section 2.1 and Table 1, Kaplan, Jared; McCandlish, Sam; Henighan, Tom; Brown, Tom B.; Chess, Benjamin; Child, Rewon; Gray, Scott; Radford, Alec; Wu, Jeffrey; Amodei, Dario (2020). "Scaling Laws for Neural Language Models". arXiv:2001.08361 [cs.LG].
44. ^ Gao, Luyu; Madaan, Aman; Zhou, Shuyan; Alon, Uri; Liu, Pengfei; Yang, Yiming; Callan, Jamie; Neubig, Graham (2022-11-01). "PAL: Program-aided Language Models". arXiv:2211.10435 [cs.CL].
45. ^ "PAL: Program-aided Language Models". *reasonwithpal.com*. Retrieved 2023-06-12.
46. ^ Paranjape, Bhargavi; Lundberg, Scott; Singh, Sameer; Hajishirzi, Hannaneh; Zettlemoyer, Luke; Tulio Ribeiro, Marco (2023-03-01). "ART: Automatic multi-step reasoning and tool-use for large language models". arXiv:2303.09014 [cs.CL].
47. ^ Liang, Yaobo; Wu, Chenfei; Song, Ting; Wu, Wenshan; Xia, Yan; Liu, Yu; Ou, Yang; Lu, Shuai; Ji, Lei; Mao, Shaoguang; Wang, Yun; Shou, Linjun; Gong, Ming; Duan, Nan (2023-03-01). "TaskMatrix.AI: Completing Tasks by Connecting Foundation Models with Millions of APIs". arXiv:2303.16434 [cs.AI].
48. ^ Patil, Shishir G.; Zhang, Tianjun; Wang, Xin; Gonzalez, Joseph E. (2023-05-01). "Gorilla: Large Language Model Connected with Massive APIs". arXiv:2305.15334 [cs.CL].
49. ^ Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman; Küttler, Heinrich; Lewis, Mike; Yih, Wen-tau; Rocktäschel, Tim; Riedel, Sebastian; Kiela, Douwe (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **33**: 9459–9474. arXiv:2005.11401.
50. ^ Huang, Wenlong; Abbeel, Pieter; Pathak, Deepak; Mordatch, Igor (2022-06-28). "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents". *Proceedings of the 39th International Conference on Machine Learning*. PMLR: 9118–9147. arXiv:2201.07207.
51. ^ Yao, Shunyu; Zhao, Jeffrey; Yu, Dian; Du, Nan; Shafran, Izhak; Narasimhan, Karthik; Cao, Yuan (2022-10-01). "ReAct: Synergizing Reasoning and Acting in Language Models". arXiv:2210.03629 [cs.CL].
52. ^ Wu, Yue; Prabhumoye, Shrimai; Min, So Yeon (24 May 2023). "SPRING: GPT-4 Out-performs RL Algorithms by Studying Papers and Reasoning". arXiv:2305.15486 [cs.AI].
53. ^ Wang, Zihao; Cai, Shaofei; Liu, Anji; Ma, Xiaojian; Liang, Yitao (2023-02-03). "Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents". arXiv:2302.01560 [cs.AI].
54. ^ Shinn, Noah; Cassano, Federico; Labash, Beck; Gopinath, Ashwin; Narasimhan, Karthik; Yao, Shunyu (2023-03-01). "Reflexion: Language Agents with Verbal Reinforcement Learning". arXiv:2303.11366 [cs.AI].
55. ^ Hao, Shibo; Gu, Yi; Ma, Haodi; Jiahua Hong, Joshua; Wang, Zhen; Zhe Wang, Daisy; Hu, Zhiting (2023-05-01). "Reasoning with Language Model is Planning with World Model". arXiv:2305.14992 [cs.CL].
56. ^ Zhang, Jenny; Lehman, Joel; Stanley, Kenneth; Clune, Jeff (2 June 2023). "OMNI: Open-endedness via Models of human Notions of Interestingness". arXiv:2306.01711 [cs.AI].
57. ^ *a b* "Voyager | An Open-Ended Embodied Agent with Large Language Models". *voyager.minedojo.org*. Retrieved 2023-06-09.
58. ^ Park, Joon Sung; O'Brien, Joseph C.; Cai, Carrie J.; Ringel Morris, Meredith; Liang, Percy; Bernstein, Michael S. (2023-04-01). "Generative Agents: Interactive Simulacra of Human Behavior". arXiv:2304.03442 [cs.HC].
59. ^ Nagel, Markus; Amjad, Rana Ali; Baalen, Mart Van; Louizos, Christos; Blankevoort, Tijmen (2020-11-21). "Up or Down? Adaptive Rounding for Post-Training Quantization". *Proceedings of the 37th International Conference on Machine Learning*. PMLR: 7197–7206.
60. ^ Polino, Antonio; Pascanu, Razvan; Alistarh, Dan (2018-02-01). "Model compression via distillation and quantization". arXiv:1802.05668 [cs.NE].

62. ^ Dettmers, Tim; Svirschevski, Ruslan; Egiazarian, Vage; Kuznedelev, Denis; Frantar, Elias; Ashkboos, Saleh; Borzunov, Alexander; Hoefler, Torsten; Alistarh, Dan (2023-06-01). "SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression". arXiv:2306.03078 [cs.CL].
63. ^ Dettmers, Tim; Pagnoni, Artidoro; Holtzman, Ari; Zettlemoyer, Luke (2023-05-01). "QLoRA: Efficient Finetuning of Quantized LLMs". arXiv:2305.14314 [cs.LG].
64. ^ Kiros, Ryan; Salakhutdinov, Ruslan; Zemel, Rich (2014-06-18). "Multimodal Neural Language Models". *Proceedings of the 31st International Conference on Machine Learning*. PMLR: 595–603.
65. ^ Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E (2012). "ImageNet Classification with Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems*. Curran Associates, Inc. **25**.
66. ^ Antol, Stanislaw; Agrawal, Aishwarya; Lu, Jiasen; Mitchell, Margaret; Batra, Dhruv; Zitnick, C. Lawrence; Parikh, Devi (2015). "VQA: Visual Question Answering". *ICCV*: 2425–2433.
67. ^ Li, Junnan; Li, Dongxu; Savarese, Silvio; Hoi, Steven (2023-01-01). "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models". arXiv:2301.12597 [cs.CV].
68. ^ Alayrac, Jean-Baptiste; Donahue, Jeff; Luc, Pauline; Miech, Antoine; Barr, Iain; Hasson, Yana; Lenc, Karel; Mensch, Arthur; Millican, Katherine; Reynolds, Malcolm; Ring, Roman; Rutherford, Eliza; Cabi, Serkan; Han, Tengda; Gong, Zhitao (2022-12-06). "Flamingo: a Visual Language Model for Few-Shot Learning". *Advances in Neural Information Processing Systems*. **35**: 23716–23736. arXiv:2204.14198.
69. ^ Driess, Danny; Xia, Fei; Sajjadi, Mehdi S. M.; Lynch, Corey; Chowdhery, Aakanksha; Ichter, Brian; Wahid, Ayzaan; Tompson, Jonathan; Vuong, Quan; Yu, Tianhe; Huang, Wenlong; Chebotar, Yevgen; Sermanet, Pierre; Duckworth, Daniel; Levine, Sergey (2023-03-01). "PaLM-E: An Embodied Multimodal Language Model". arXiv:2303.03378 [cs.LG].
70. ^ Liu, Haotian; Li, Chunyuan; Wu, Qingyang; Lee, Yong Jae (2023-04-01). "Visual Instruction Tuning". arXiv:2304.08485 [cs.CV].
71. ^ Zhang, Hang; Li, Xin; Bing, Lidong (2023-06-01). "Video-LLaMA: An Instruction-tuned Audio-Visual Language Model for Video Understanding". arXiv:2306.02858 [cs.CL].
72. ^ OpenAI (2023-03-27). "GPT-4 Technical Report". arXiv:2303.08774 [cs.CL].
73. ^ OpenAI (September 25, 2023). "GPT-4V(ision) System Card" (PDF).
74. ^ Pichai, Sundar, *Google Keynote (Google I/O '23)*, timestamp 15:31, retrieved 2023-07-02
75. ^ Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; Buchatskaya, Elena; Cai, Trevor; Rutherford, Eliza; Casas, Diego de Las; Hendricks, Lisa Anne; Welbl, Johannes; Clark, Aidan; Hennigan, Tom; Noland, Eric; Millican, Katie; Driessche, George van den; Damoc, Bogdan (2022-03-29). "Training Compute-Optimal Large Language Models". arXiv:2203.15556 [cs.CL].
76. ^ *a b* Caballero, Ethan; Gupta, Kshitij; Rish, Irina; Krueger, David (2022). "Broken Neural Scaling Laws". arXiv:2210.14891 [cs.LG].
77. ^ "137 emergent abilities of large language models". *Jason Wei*. Retrieved 2023-06-24.
78. ^ Hahn, Michael; Goyal, Navin (2023-03-14). "A Theory of Emergent In-Context Learning as Implicit Structure Induction". arXiv:2303.07971 [cs.LG].
79. ^ Pilehvar, Mohammad Taher; Camacho-Collados, Jose (June 2019). "Proceedings of the 2019 Conference of the North". *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics: 1267–1273. doi:10.18653/v1/N19-1128. S2CID 102353817.
80. ^ "WiC: The Word-in-Context Dataset". *pilehvar.github.io*. Retrieved 2023-06-27.
81. ^ Patel, Roma; Pavlick, Ellie (2021-10-06). "Mapping Language Models to Grounded Conceptual Spaces". *ICLR*.
82. ^ *A Closer Look at Large Language Models Emergent Abilities* (Yao Fu, Nov 20, 2022)
83. ^ Ornes, Stephen (March 16, 2023). "The Unpredictable Abilities Emerging From Large AI Models". *Quanta Magazine*.
84. ^ Schaeffer, Rylan; Miranda, Brando; Koyejo, Sanmi (2023-04-01). "Are Emergent Abilities of Large Language Models a Mirage?". arXiv:2304.15004 [cs.AI].
85. ^ Li, Kenneth; Hopkins, Aspen K.; Bau, David; Viégas, Fernanda; Pfister, Hanspeter; Wattenberg, Martin (2022-10-01). "Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task". arXiv:2210.13382 [cs.LG].
86. ^ "Large Language Model: world models or surface statistics?". *The Gradient*. 2023-01-21. Retrieved 2023-06-12.
87. ^ Jin, Charles; Rinard, Martin (2023-05-01). "Evidence of Meaning in Language Models Trained on Programs". arXiv:2305.11169 [cs.LG].
88. ^ Nanda, Neel; Chan, Lawrence; Lieberum, Tom; Smith, Jess; Steinhardt, Jacob (2023-01-01). "Progress measures for grokking via mechanistic interpretability". arXiv:2301.05217 [cs.LG].
89. ^ *a b c d e* Mitchell, Melanie; Krakauer, David C. (28 March 2023). "The debate over understanding in AI's large language models". *Proceedings of the National Academy of Sciences*. **120** (13): e2215907120. arXiv:2210.13966. Bibcode:2023PNAS..12015907M. doi:10.1073/pnas.2215907120. PMC 10068812. PMID 36943882.
90. ^ Metz, Cade (16 May 2023). "Microsoft Says New A.I. Shows Signs of Human Reasoning". *The New York Times*.
91. ^ *a b* Bubeck, Sébastien; Chandrasekaran, Varun; Eldan, Ronen; Gehrke, Johannes; Horvitz, Eric; Kamar, Ece; Lee, Peter; Lee, Yin Tat; Li, Yuanzhi; Lundberg, Scott; Nori, Harsha; Palangi, Hamid; Ribeiro, Marco Tulio; Zhang, Yi (2023). "Sparks of Artificial General Intelligence: Early experiments with GPT-4". arXiv:2303.12712 [cs.CL].
92. ^ "ChatGPT is more like an 'alien intelligence' than a human brain, says futurist". *ZDNET*. 2023. Retrieved 12 June 2023.
93. ^ *a b* Newport, Cal (13 April 2023). "What Kind of Mind Does ChatGPT Have?". *The New Yorker*. Retrieved 12 June 2023.
94. ^ Roose, Kevin (30 May 2023). "Why an Octopus-like Creature Has Come to Symbolize the State of A.I." *The New York Times*. Retrieved 12 June 2023.
95. ^ "The A to Z of Artificial Intelligence". *Time Magazine*. 13 April 2023. Retrieved 12 June 2023.
96. ^ Ji, Ziwei; Lee, Nayeon; Frieske, Rita; Yu, Tiezheng; Su, Dan; Xu, Yan; Ishii, Etsuko; Bang, Yejin; Dai, Wenliang; Madotto, Andrea; Fung, Pascale (November 2022). "Survey of Hallucination in Natural Language Generation" (pdf). *ACM Computing Surveys*. Association for Computing Machinery. **55** (12): 1–38. arXiv:2202.03629. doi:10.1145/3571730. S2CID 246652372. Retrieved 15 January 2023.
97. ^ Varshney, Neeraj; Yao, Wenlin; Zhang, Hongming; Chen, Jianshu; Yu, Dong (2023). "A Stitch in Time Saves Nine: Detecting and Mitigating Hallucinations of LLMs by Validating Low-Confidence Generation". arXiv:2307.03987 [cs.CL].
98. ^ Lakoff, George (1999). *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Philosophy; Appendix: The Neural Theory of Language Paradigm*. New York Basic Books. pp. 569–583. ISBN 978-0-465-05674-3.
99. ^ Evans, Vyvyan. (2014). *The Language Myth*. Cambridge University Press. ISBN 978-1-107-04396-1.
100. ^ Friston, Karl J. (2022). *Active Inference: The Free Energy Principle in Mind, Brain, and Behavior; Chapter 4 The Generative Models of Active Inference*. The MIT Press. ISBN 978-0-262-36997-8.
101. ^ Huyen, Chip (2019). "Understanding Evaluation Metrics for Language Modeling". The Gradient. Retrieved January 14, 2024.
102. ^ *a b* Clark, Christopher; Lee, Kenton; Chang, Ming-Wei; Kwiatkowski, Tom; Collins, Michael; Toutanova, Kristina (2019). "BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions". arXiv:1905.10044 [cs.CL].
103. ^ *a b c* Wayne Xin Zhao; Zhou, Kun; Li, Junyi; Tang, Tianyi; Wang, Xiaolei; Hou, Yupeng; Min, Yingqian; Zhang, Beichen; Zhang, Junjie; Dong, Zican; Du, Yifan; Yang, Chen; Chen, Yushuo; Chen, Zhipeng; Jiang, Jinhao; Ren, Ruiyang; Li, Yifan; Tang, Xinyu; Liu, Zikang; Liu, Peiyu; Nie, Jian-Yun; Wen, Ji-Rong (2023). "A Survey of Large Language Models". arXiv:2303.18223 [cs.CL].
104. ^ Huyen, Chip (18 October 2019). "Evaluation Metrics for Language Modeling". *The Gradient*.
105. ^ Srivastava, Aarohi; et al. (2022). "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models". arXiv:2206.04615 [cs.CL].
106. ^ Lin, Stephanie; Hilton, Jacob; Evans, Owain (2021). "TruthfulQA:

61. ^ Frantar, Elias; Ashkboos, Saleh; Hoefler, Torsten; Alistarh, Dan (2022-10-01). "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers". arXiv:2210.17323 [cs.LG].

Measuring How Models Mimic Human Falsehoods". arXiv:2109.07958 [cs.CL].

107. ^ a b Zellers, Rowan; Holtzman, Ari; Bisk, Yonatan; Farhadi, Ali; Choi, Yejin (2019). "HellaSwag: Can a Machine Really Finish Your Sentence?". arXiv:1905.07830 [cs.CL].

108. ^ "Prepare for truly useful large language models". *Nature Biomedical Engineering*. **7** (2): 85–86. 7 March 2023. doi:10.1038/s41551-023-01012-6. PMID 36882584. S2CID 257403466.
109. ^ "Your job is (probably) safe from artificial intelligence". *The Economist*. 7 May 2023. Retrieved 18 June 2023.
110. ^ "Generative AI Could Raise Global GDP by 7%". *Goldman Sachs*. Retrieved 18 June 2023.
111. ^ Peng, Zhencan; Wang, Zhizhi; Deng, Dong (13 June 2023). "Near-Duplicate Sequence Search at Scale for Large Language Model Memorization Evaluation" (PDF). *Proceedings of the ACM on Management of Data*. **1** (2): 1–18. doi:10.1145/3589324. S2CID 259213212. Retrieved 2024-01-20. Citing Lee et al 2022.
112. ^ Peng, Wang & Deng 2023, p. 8.
113. ^ Alba, Davey (1 May 2023). "AI chatbots have been used to create dozens of news content farms". *The Japan Times*. Retrieved 18 June 2023.
114. ^ "Could chatbots help devise the next pandemic virus?". *Science*. 14 June 2023. doi:10.1126/science.adj2463.
115. ^ Stephen Council (1 Dec 2023). "How Googlers cracked an SF rival's tech model with a single word". SFGATE.
116. ^ Hubinger, Evan (10 January 2024). "Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training". arXiv:2401.05566 [cs.CR].
117. ^ a b Stokel-Walker, Chris (November 22, 2023). "ChatGPT Replicates Gender Bias in Recommendation Letters". *Scientific American*. Retrieved 2023-12-29.
118. ^ Luo, Queenie; Puett, Michael J.; Smith, Michael D. (2023-03-28). "A Perspectival Mirror of the Elephant: Investigating Language Bias on Google, ChatGPT, Wikipedia, and YouTube". arXiv:2303.16281v2 [cs.CY].
119. ^ Cheng, Myra; Durmus, Esin; Jurafsky, Dan (2023-05-29), *Marked Personas: Using Natural Language Prompts to Measure Stereotypes in Language Models*, arXiv:2305.18189
120. ^ Kotek, Hadas; Dockum, Rikker; Sun, David (2023-11-05). "Gender bias and stereotypes in Large Language Models". *Proceedings of the ACM Collective Intelligence Conference*. CI '23. New York, NY, USA: Association for Computing Machinery. pp. 12–24. doi:10.1145/3582269.3615599. ISBN 979-8-4007-0113-9.
121. ^ Heikkilä, Melissa (August 7, 2023). "AI language models are rife with different political biases". *MIT Technology Review*. Retrieved 2023-12-29.
122. ^ "finetune-transformer-lm". *GitHub*. Retrieved 2 January 2024.
123. ^ a b Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2 [cs.CL].
124. ^ Prickett, Nicole Hemsoth (2021-08-24). "Cerebras Shifts Architecture To Meet Massive AI/ML Models". *The Next Platform*. Retrieved 2023-06-20.
125. ^ "BERT". March 13, 2023 – via GitHub.
126. ^ Patel, Ajay; Li, Bryan; Rasooli, Mohammad Sadegh; Constant, Noah; Raffel, Colin; Callison-Burch, Chris (2022). "Bidirectional Language Models Are Also Few-shot Learners". arXiv:2209.14500 [cs.LG].
127. ^ "BERT, RoBERTa, DistilBERT, XLNet: Which one to use?". *KDnuggets*.
128. ^ "xlnet". *GitHub*. Retrieved 2 January 2024.
129. ^ Naik, Amit Raja (September 23, 2021). "Google Introduces New Architecture To Reduce Cost Of Transformers". *Analytics India Magazine*.
130. ^ Yang, Zhilin; Dai, Zihang; Yang, Yiming; Carbonell, Jaime; Salakhutdinov, Ruslan; Le, Quoc V. (2 January 2020). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". arXiv:1906.08237 [cs.CL].
131. ^ "GPT-2: 1.5B Release". *OpenAI*. 2019-11-05. Archived from the original on 2019-11-14. Retrieved 2019-11-14.
132. ^ "Better language models and their implications". *openai.com*.
133. ^ a b "OpenAI's GPT-3 Language Model: A Technical Overview". *lambdalabs.com*. 3 June 2020.
134. ^ "gpt-2". *GitHub*. Retrieved 13 March 2023.
135. ^ Table D.1 in Brown, Tom B.; Mann, Benjamin; Ryder, Nick; Subbiah, Melanie; Kaplan, Jared; Dhariwal, Prafulla; Neelakantan, Arvind; Shyam, Pranav; Sastry, Girish; Askell, Amanda; Agarwal, Sandhini; Herbert-Voss, Ariel; Krueger, Gretchen; Henighan, Tom; Child, Rewon; Ramesh, Aditya; Ziegler, Daniel M.; Wu, Jeffrey; Winter, Clemens; Hesse, Christopher; Chen, Mark; Sigler, Eric; Litwin, Mateusz; Gray, Scott; Chess, Benjamin; Clark, Jack; Berner, Christopher; McCandlish, Sam; Radford, Alec; Sutskever, Ilya; Amodei, Dario (May 28, 2020). "Language Models are Few-Shot Learners". arXiv:2005.14165v4 [cs.CL].
136. ^ "ChatGPT: Optimizing Language Models for Dialogue". *OpenAI*. 2022-11-30. Retrieved 2023-01-13.
137. ^ "GPT Neo". March 15, 2023 – via GitHub.
138. ^ a b c Gao, Leo; Biderman, Stella; Black, Sid; Golding, Laurence; Hoppe, Travis; Foster, Charles; Phang, Jason; He, Horace; Thite, Anish; Nabeshima, Noa; Presser, Shawn; Leahy, Connor (31 December 2020). "The Pile: An 800GB Dataset of Diverse Text for Language Modeling". arXiv:2101.00027 [cs.CL].
139. ^ a b Iyer, Abhishek (15 May 2021). "GPT-3's free alternative GPT-Neo is something to be excited about". *VentureBeat*.
140. ^ "GPT-J-6B: An Introduction to the Largest Open Source GPT Model | Forefront". *www.forefront.ai*. Retrieved 2023-02-28.
141. ^ a b c d Dey, Nolan; Gosal, Gurpreet; Zhiming; Chen; Khachane, Hemant; Marshall, William; Pathria, Ribhu; Tom, Marvin; Hestness, Joel (2023-04-01). "Cerebras-GPT: Open Compute-Optimal Language Models Trained on the Cerebras Wafer-Scale Cluster". arXiv:2304.03208 [cs.LG].
142. ^ Alvi, Ali; Kharya, Paresh (11 October 2021). "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model". *Microsoft Research*.
143. ^ a b Smith, Shaden; Patwary, Mostofa; Norick, Brandon; LeGresley, Patrick; Rajbhandari, Samyam; Casper, Jared; Liu, Zhun; Prabhumoye, Shrimai; Zerveas, George; Korthikanti, Vijay; Zhang, Elton; Child, Rewon; Aminabadi, Reza Yazdani; Bernauer, Julie; Song, Xia (2022-02-04). "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model". arXiv:2201.11990 [cs.CL].
144. ^ Wang, Shuohuan; Sun, Yu; Xiang, Yang; Wu, Zhihua; Ding, Siyu; Gong, Weibao; Feng, Shikun; Shang, Junyuan; Zhao, Yanbin; Pang, Chao; Liu, Jiaxiang; Chen, Xuyi; Lu, Yuxiang; Liu, Weixin; Wang, Xi; Bai, Yangfan; Chen, Qiuliang; Zhao, Li; Li, Shiyong; Sun, Peng; Yu, Dianhai; Ma, Yanjun; Tian, Hao; Wu, Hua; Wu, Tian; Zeng, Wei; Li, Ge; Gao, Wen; Wang, Haifeng (December 23, 2021). "ERNIE 3.0 Titan: Exploring Larger-scale Knowledge Enhanced Pre-training for Language Understanding and Generation". arXiv:2112.12731 [cs.CL].
145. ^ "Product". *Anthropic*. Retrieved 14 March 2023.
146. ^ a b Askell, Amanda; Bai, Yuntao; Chen, Anna; et al. (9 December 2021). "A General Language Assistant as a Laboratory for Alignment". arXiv:2112.00861 [cs.CL].
147. ^ Bai, Yuntao; Kadavath, Saurav; Kundu, Sandipan; et al. (15 December 2022). "Constitutional AI: Harmlessness from AI Feedback". arXiv:2212.08073 [cs.CL].
148. ^ "Language modelling at scale: Gopher, ethical considerations, and retrieval". *www.deepmind.com*. 8 December 2021. Retrieved 20 March 2023.
149. ^ a b c Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; et al. (29 March 2022). "Training Compute-Optimal Large Language Models". arXiv:2203.15556 [cs.CL].
150. ^ a b c d Table 20 and page 66 of *PaLM: Scaling Language Modeling with Pathways*
151. ^ a b Cheng, Heng-Tze; Thoppilan, Romal (January 21, 2022). "LaMDA: Towards Safe, Grounded, and High-Quality Dialog Models for Everything". *ai.googleblog.com*. Retrieved 2023-03-09.
152. ^ Thoppilan, Romal; De Freitas, Daniel; Hall, Jamie; Shazeer, Noam; Kulshreshtha, Apoorv; Cheng, Heng-Tze; Jin, Alicia; Bos, Taylor; Baker, Leslie; Du, Yu; Li, YaGuang; Lee, Hongrae; Zheng, Huaixiu Steven; Ghafouri, Amin; Menegali, Marcelo (2022-01-01). "LaMDA: Language Models for Dialog Applications". arXiv:2201.08239 [cs.CL].
153. ^ Black, Sidney; Biderman, Stella; Hallahan, Eric; et al. (2022-05-01). *GPT-NeoX-20B: An Open-Source Autoregressive Language Model*. Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models. Vol. Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models. pp. 95–136. Retrieved 2022-12-19.
154. ^ a b c Hoffmann, Jordan; Borgeaud, Sebastian; Mensch, Arthur; Sifre, Laurent (12 April 2022). "An empirical analysis of compute-optimal large language model training". *Deepmind Blog*.
155. ^ Narang, Sharan; Chowdhery, Aakanksha (April 4, 2022). "Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance". *ai.googleblog.com*. Retrieved 2023-03-09.
156. ^ "Democratizing access to large-scale language models with OPT-175B". *ai.facebook.com*.
157. ^ Zhang, Susan; Roller, Stephen; Goyal, Naman; Artetxe, Mikel; Chen, Moya; Chen, Shuohui; Dewan, Christopher; Diab, Mona; Li, Xian; Lin, Xi Victoria; Mihaylov, Todor; Ott, Myle; Shleifer, Sam; Shuster, Kurt; Simig, Daniel; Koura, Punit Singh; Sridhar, Anjali; Wang, Tianlu; Zettlemoyer, Luke (21 June 2022). "OPT: Open Pre-trained Transformer Language Models". arXiv:2205.01068 [cs.CL].
158. ^ a b Khrushchev, Mikhail; Vasilev, Ruslan; Petrov, Alexey; Zinov, Nikolay (2022-06-22), *YaLM 100B*, retrieved 2023-03-18
159. ^ a b Lewkowycz, Aitor; Andreassen, Anders; Dohan, David; Dyer, Ethan; Michalewski, Henryk; Ramasesh, Vinay; Slone, Ambrose; Anil, Cem; Schlag, Imanol; Gutman-Solo, Theo; Wu, Yuhuai; Neyshabur, Behnam; Gur-Ari, Guy; Misra, Vedant (30 June 2022). "Solving Quantitative Reasoning Problems with Language Models". arXiv:2206.14858 [cs.CL].
160. ^ "Minerva: Solving Quantitative Reasoning Problems with Language Models". *ai.googleblog.com*. 30 June 2022. Retrieved 20 March 2023.
161. ^ Ananthaswamy, Anil (8 March 2023). "In AI, is bigger always better?". *Nature*. **615** (7951): 202–205. Bibcode:2023Natur.615..202A. doi:10.1038/d41586-023-00641-w. PMID 36890378. S2CID 257380916.
162. ^ "bigscience/bloom · Hugging Face". *huggingface.co*.
163. ^ Taylor, Ross; Kardas, Marcin; Cucurull, Guillem; Scialom, Thomas; Hartshorn, Anthony; Saravia, Elvis; Poulton, Andrew; Kerkez, Viktor; Stojnic, Robert (16 November 2022). "Galactica: A Large Language Model for Science". arXiv:2211.09085 [cs.CL].
164. ^ "20B-parameter Alexa model sets new marks in few-shot learning". *Amazon Science*. 2 August 2022.
165. ^ Soltan, Saleh; Ananthakrishnan, Shankar; FitzGerald, Jack; et al. (3 August 2022). "AlexaTM 20B: Few-Shot Learning Using a Large-Scale Multilingual Seq2Seq Model". arXiv:2208.01448 [cs.CL].
166. ^ "AlexaTM 20B is now available in Amazon SageMaker JumpStart | AWS Machine Learning Blog". *aws.amazon.com*. 17 November 2022. Retrieved 13 March 2023.
167. ^ a b c "Introducing LLaMA: A foundational, 65-billion-parameter large language model". *Meta AI*. 24 February 2023.
168. ^ a b c "The Falcon has landed in the Hugging Face ecosystem". *huggingface.co*. Retrieved 2023-06-20.
169. ^ "Stanford CRFM". *crfm.stanford.edu*.
170. ^ "GPT-4 Technical Report" (PDF). *OpenAI*. 2023. Archived (PDF) from the original on March 14, 2023. Retrieved March 14, 2023.
171. ^ Dey, Nolan (March 28, 2023). "Cerebras-GPT: A Family of Open, Compute-efficient, Large Language Models". *Cerebras*.
172. ^ "Abu Dhabi-based TII launches its own version of ChatGPT". *tii.ae*.
173. ^ Penedo, Guilherme; Malartic, Quentin; Hesslow, Daniel; Cojocaru, Ruxandra; Cappelli, Alessandro; Alobeidli, Hamza; Pannier, Baptiste; Almazrouei, Ebtesam; Launay, Julien (2023-06-01). "The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only". arXiv:2306.01116 [cs.CL].
174. ^ "tiiuae/falcon-40b · Hugging Face". *huggingface.co*. 2023-06-09. Retrieved 2023-06-20.
175. ^ UAE's Falcon 40B, World's Top-Ranked AI Model from Technology Innovation Institute, is Now Royalty-Free, 31 May 2023
176. ^ Wu, Shijie; Irsoy, Ozan; Lu, Steven; Dabravolski, Vadim; Dredze, Mark; Gehrmann, Sebastian; Kambadur, Prabhanjan; Rosenberg, David; Mann, Gideon (March 30, 2023). "BloombergGPT: A Large Language Model for Finance". arXiv:2303.17564 [cs.LG].
177. ^ Ren, Xiaozhe; Zhou, Pingyi; Meng, Xinfan; Huang, Xinjing; Wang, Yadao; Wang, Weichao; Li, Pengfei; Zhang, Xiaoda; Podolskiy, Alexander; Arshinov, Grigory; Bout, Andrey; Piontkovskaya, Irina; Wei, Jiansheng; Jiang, Xin; Su, Teng; Liu, Qun; Yao, Jun (March 19, 2023). "PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing". arXiv:2303.10845 [cs.CL].
178. ^ Köpf, Andreas; Kilcher, Yannic; von Rütte, Dimitri; Anagnostidis, Sotiris; Tam, Zhi-Rui; Stevens, Keith; Barhoum, Abdullah; Duc, Nguyen Minh; Stanley, Oliver; Nagyfi, Richárd; ES, Shahul; Suri, Sameer; Glushkov, David; Dantuluri, Arnav; Maguire, Andrew (2023-04-14). "OpenAssistant Conversations – Democratizing Large Language Model Alignment". arXiv:2304.07327 [cs.CL].
179. ^ Wrobel, Sharon. "Tel Aviv startup rolls out new advanced AI language model to rival OpenAI". *www.timesofisrael.com*. Retrieved 2023-07-24.
180. ^ Wiggers, Kyle (2023-04-13). "With Bedrock, Amazon enters the generative AI race". *TechCrunch*. Retrieved 2023-07-24.
181. ^ a b Elias, Jennifer (16 May 2023). "Google's newest A.I. model uses nearly five times more text data for training than its predecessor". *CNBC*. Retrieved 18 May 2023.
182. ^ "Introducing PaLM 2". *Google*. May 10, 2023.
183. ^ a b "Introducing Llama 2: The Next Generation of Our Open Source Large Language Model". *Meta AI*. 2023. Retrieved 2023-07-19.
184. ^ "Claude 2". *anthropic.com*. Retrieved 12 December 2023.
185. ^ a b "Falcon 180B". *Technology Innovation Institute*. 2023. Retrieved 2023-09-21.
186. ^ "Announcing Mistral 7B". *Mistral*. 2023. Retrieved 2023-10-06.
187. ^ "Introducing Claude 2.1". *anthropic.com*. Retrieved 12 December 2023.
188. ^ "Grok-1 model card". *x.ai*. Retrieved 12 December 2023.
189. ^ "Gemini – Google DeepMind". *deepmind.google*. Retrieved 12 December 2023.
190. ^ "Mixtral of experts". *mistral.ai*. 11 December 2023. Retrieved 12 December 2023.
191. ^ Franzen, Carl (11 December 2023). "Mistral shocks AI community as latest open source model eclipses GPT-3.5 performance". *VentureBeat*. Retrieved 12 December 2023.
192. ^ Hughes, Alyssa (12 December 2023). "Phi-2: The surprising power of small language models". *Microsoft Research*. Retrieved 13 December 2023.
193. ^ Cheah, Eugene. "□ Eagle 7B : Soaring past Transformers with 1 Trillion Tokens Across 100+ Languages (RWKV-v5)". *blog.rwkv.com*. Retrieved 31 January 2024.
194. ^ "Our next-generation model: Gemini 1.5". *Google*. 15 February 2024. Retrieved 16 February 2024. "This means 1.5 Pro can process vast amounts of information in one go — including 1 hour of video, 11 hours of audio, codebases with over 30,000 lines of code or over 700,000 words. In our research, we've also successfully tested up to 10 million tokens."
195. ^ "Gemma" – via GitHub.
196. ^ "Introducing the next generation of Claude". *www.anthropic.com*. Retrieved 2024-03-04.

Further reading [edit]

- Jurafsky, Dan, Martin, James. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd Edition draft, 2023.
- Phuong, Mary; Hutter, Marcus (2022). "Formal Algorithms for Transformers". arXiv:2207.09238 [cs.LG].
- Eloundou, Tyna; Manning, Sam; Mishkin, Pamela; Rock, Daniel (2023). "GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models". arXiv:2303.10130 [econ.GN].
- Eldan, Ronen; Li, Yuanzhi (2023). "TinyStories: How Small Can Language Models Be and Still Speak Coherent English?". arXiv:2305.07759 [cs.CL].
- Frank, Michael C. (27 June 2023). "Baby steps in evaluating the capacities of large language models". *Nature Reviews Psychology*. **2** (8): 451–452. doi:10.1038/s44159-023-00211-x. ISSN 2731-0574. S2CID 259713140. Retrieved 2 July 2023.

- Zhao, Wayne Xin; et al. (2023). "A Survey of Large Language Models". arXiv:2303.18223 [cs.CL].
- Kaddour, Jean; et al. (2023). "Challenges and Applications of Large Language Models". arXiv:2307.10169 [cs.CL].
- Yin, Shukang; Fu, Chaoyou; Zhao, Sirui; Li, Ke; Sun, Xing; Xu, Tong; Chen, Enhong (2023-06-01). "A Survey on Multimodal Large Language Models". arXiv:2306.13549 [cs.CV].

- Open LLMs repository on GitHub.

V • T • E Natural language processing [show]

Categories: Large language models | Deep learning | Natural language processing

This page was last edited on 12 March 2024, at 21:05 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License 4.0; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Code of Conduct Developers Statistics Cookie statement Mobile view