Exhibit 51

Claude Docs

Resources > Glossary

Glossary

These concepts are not unique to Anthropic's language models, but we present a brief summary of key terms below.

Copy page

Context window

The "context window" refers to the amount of text a language model can look back on and reference when generating new text. This is different from the large corpus of data the language model was trained on, and instead represents a "working memory" for the model. A larger context window allows the model to understand and respond to more complex and lengthy prompts, while a smaller context window may limit the model's ability to handle longer prompts or maintain coherence over extended conversations.

See our guide to understanding context windows to learn more.

Fine-tuning

Fine-tuning is the process of further training a pretrained language model using additional data. This causes the model to start representing and mimicking the patterns and characteristics of the fine-tuning dataset. Claude is not a bare language model; it has already been fine-tuned to be a helpful assistant. Our API does not currently offer fine-tuning, but please ask your Anthropic contact if you are interested in exploring this option. Fine-tuning can be useful for adapting a language model to a specific domain, task, or writing style, but it requires careful consideration of the fine-tuning data and the potential impact on the model's performance and biases.

HHH

These three H's represent Anthropic's goals in ensuring that Claude is b Ask Docs
society:

- An **honest** AI will give accurate information, and not hallucinate or confabulate. It will acknowledge its limitations and uncertainties when appropriate.
- A **harmless** AI will not be offensive or discriminatory, and when asked to aid in a dangerous or unethical act, the AI should politely refuse and explain why it cannot comply.

Latency

Latency, in the context of generative AI and large language models, refers to the time it takes for the model to respond to a given prompt. It is the delay between submitting a prompt and receiving the generated output. Lower latency indicates faster response times, which is crucial for real-time applications, chatbots, and interactive experiences. Factors that can affect latency include model size, hardware capabilities, network conditions, and the complexity of the prompt and the generated response.

LLM

Large language models (LLMs) are AI language models with many parameters that are capable of performing a variety of surprisingly useful tasks. These models are trained on vast amounts of text data and can generate human-like text, answer questions, summarize information, and more. Claude is a conversational assistant based on a large language model that has been fine-tuned and trained using RLHF to be more helpful, honest, and harmless.

MCP (Model Context Protocol)

Model Context Protocol (MCP) is an open protocol that standardizes how applications provide context to LLMs. Like a USB-C port for AI applications, MCP provides a unified way to connect AI models to different data sources and tools. MCP enables AI systems to maintain consistent context across interactions and access external resources in a standardized manner. See our MCP documentation to learn more.

MCP connector

The MCP connector is a feature that allows API users to connect to MCP Ask Docs from the Messages API without building an MCP client. This enables seamless integration

documentation to learn more.

Pretraining

Pretraining is the initial process of training language models on a large unlabeled corpus of text. In Claude's case, autoregressive language models (like Claude's underlying model) are pretrained to predict the next word, given the previous context of text in the document. These pretrained models are not inherently good at answering questions or following instructions, and often require deep skill in prompt engineering to elicit desired behaviors. Fine-tuning and RLHF are used to refine these pretrained models, making them more useful for a wide range of tasks.

RAG (Retrieval augmented generation)

Retrieval augmented generation (RAG) is a technique that combines information retrieval with language model generation to improve the accuracy and relevance of the generated text, and to better ground the model's response in evidence. In RAG, a language model is augmented with an external knowledge base or a set of documents that is passed into the context window. The data is retrieved at run time when a query is sent to the model, although the model itself does not necessarily retrieve the data (but can with tool use and a retrieval function). When generating text, relevant information first must be retrieved from the knowledge base based on the input prompt, and then passed to the model along with the original query. The model uses this information to guide the output it generates. This allows the model to access and utilize information beyond its training data, reducing the reliance on memorization and improving the factual accuracy of the generated text. RAG can be particularly useful for tasks that require up-to-date information, domain-specific knowledge, or explicit citation of sources. However, the effectiveness of RAG depends on the quality and relevance of the external knowledge base and the knowledge that is retrieved at runtime.

RLHF

Reinforcement Learning from Human Feedback (RLHF) is a technique used to train a pretrained language model to behave in ways that are consistent with human preferences. This can include helping the model follow instructions more act more like a chatbot. Human feedback consists of ranking a set of two texts, and the reinforcement learning process encourages the model to prefer outputs

Ask Docs

Temperature

Temperature is a parameter that controls the randomness of a model's predictions dur
text generation. Higher temperatures lead to more creative and diverse outputs, allowing for multiple variations in phrasing and, in the case of fiction, variation in answers as well. Lower temperatures result in more conservative and deterministic outputs that stick to the most probable phrasing and answers. Adjusting the temperature enables users to encourage a language model to explore rare, uncommon, or surprising word choices and sequences, rather than only selecting the most likely predictions.

TTFT (Time to first token)

Time to First Token (TTFT) is a performance metric that measures the time it takes for a language model to generate the first token of its output after receiving a prompt. It is an important indicator of the model's responsiveness and is particularly relevant for interactive applications, chatbots, and real-time systems where users expect quick initial feedback. A lower TTFT indicates that the model can start generating a response faster, providing a more seamless and engaging user experience. Factors that can influence TTFT include model size, hardware capabilities, network conditions, and the complexity of the prompt.

Tokens

Tokens are the smallest individual units of a language model, and can correspond to words, subwords, characters, or even bytes (in the case of Unicode). For Claude, a token approximately represents 3.5 English characters, though the exact number can vary depending on the language used. Tokens are typically hidden when interacting with language models at the "text" level but become relevant when examining the exact inputs and outputs of a language model. When Claude is provided with text to evaluate, the text (consisting of a series of characters) is encoded into a series of tokens for the model to process. Larger tokens enable data efficiency during inference and pretraining (and are utilized when possible), while smaller tokens allow a model to handle uncommon or never-before-seen words. The choice of tokenization method can impact the model's performance, vocabulary size, and ability to handle out-of-vocabulary wr

Ask Docs

Claude Docs

Solutions
AI agents
Code modernization
Coding
Customer support
Education
Financial services
Government
Life sciences

Partners
Amazon Bedrock
Google Cloud's Vertex AI

Learn
Blog
Catalog
Courses
Use cases
Connectors
Customer stories
Engineering at Anthropic
Events
Powered by Claude
Service partners
Startups program

Company
Anthropic
Careers
Economic Futures
Research
News
Responsible Scaling Policy
Security and compliance
Transparency

Ask Docs

Status
Support
Discord

Terms and policies
Privacy policy
Responsible disclosure policy
Terms of service: Commercial
Terms of service: Consumer
Usage policy



Ask Docs