Online glossary[1]

API (application programming interface): a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. Specific to LLMs, it is a set of protocols and tools that provide an easy and convenient way for users to interact with the model without needing extensive technical expertise or computational resources. By sending input text and receiving responses through simple API calls, users can integrate LLM capabilities into applications and workflows without managing complex infrastructure, making the model accessible and efficient for a wide range of use cases.

Artificial neural network (ANN): consists of interconnected units or nodes called artificial neurons, which are loosely inspired by biological neurons in the brain. These neurons are connected by edges that model the synapses in the brain. Each artificial neuron receives inputs from other neurons or from the system's inputs, performs a weighted sum of these inputs (vector multiplication) and then passes the output through a non-linear activation function to another neuron or to the system's output. ANNs can be mathematically represented by nested functions: $o = \sigma(W_K\sigma(W_{K-1}\sigma(\dots W_2\sigma(W_1x + b_1) + b_2) + b_{K-1}) + b_K)$, where x is a d_0-dimensional vector input, W_i is a $d_i \times d_{i-1}$ matrix, b_i d_i-dimensional vector, and $\sigma(\cdot)$ is the non-linear activation function applied element-wise. d_i is the number of hidden neurons in each layer. W_i and b_i are the trainable parameters to learn a given function. ANNs are universal approximators. They come in various forms, each defined by specific structural constraints on the parameters and non-linear functions. The most common types include fully connected neural networks (FCNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers.

BERT (bidirectional encoder representations from transformers): a popular transformer-based model that processes the context of words by looking at both sides of a word in a sentence. BERT is trained with a global objective, meaning it aims to process the entire input text. This produces an extra embedding: the CLS embedding, which represents the entire sentence or document. This global embedding is useful for solving classification or estimation questions about the input data, such as sentiment analysis. For more details, see Devlin et al (2018).

Chatbot: a fine-tuned model of a generative LLM designed to simulate human conversation in a natural and interactive way. Chatbots are trained on large amounts of conversational data and then fine-tuned to perform specific tasks.

Chain-of-thought (CoT): a reasoning process where a human breaks down a complex task into a sequence of intermediate steps to arrive at a logical conclusion to help an LLM replicate the process. Instead of jumping straight to the answer, the model analyses each component of the task methodically. For example, when classifying the monetary policy sentiment of a given sentence, the model first identifies key phrases related to economic indicators, then evaluates any mentions of risk or uncertainty, assesses the overall tone of the statement and finally determines the sentiment in the context of monetary policy. This step-by-step approach ensures a more nuanced and accurate classification.

Chunk: a group of words that together form a meaningful unit, a unit of input for an LLM to process, such as a sentence or paragraph. Chunks are used to divide larger texts into manageable segments for efficient processing and analysis.

CLS (classification token): a special token in transformer models like BERT that serves as a summarised representation of the entire input sequence.

Clustering: grouping similar data points together based on their characteristics, often used to find patterns or structures in data.

Context window: the amount of text a language model can process concurrently. It is measured in thousands of tokens and, in a new model, can go up to millions.

Deep learning: artificial neutral network with numerous layers.

[1] This online glossary accompanies the feature "Large language models: a primer for economists" by Kwon, Park, Perez-Cruz, Rungcharoenkitkul, BIS Quarterly Review, December 2024.

Dimension reduction: techniques to reduce the complexity of high-dimensional data, such as embeddings, by representing it in fewer dimensions, enhancing efficiency and performance. Common techniques include:

- **PCA (principal component analysis)**: a linear dimensionality reduction technique that transforms data by identifying principal components – directions in the vector space that maximise variance. PCA is computationally efficient and is commonly used when the goal is to simplify data while retaining as much variation as possible. But it may struggle with capturing non-linear patterns.
- **Autoencoder**: an artificial neural network designed for unsupervised learning. It consists of two main functions – an encoding function that transforms the input data into a reduced representation and a decoding function that reconstructs the input data from this encoded form. The primary goal of an autoencoder is to learn an efficient representation used for dimensionality reduction. This lower-dimensional embedding can then be utilised by other machine learning algorithms for further analysis or processing.
- **t-SNE (t-distributed stochastic neighbour embedding)**: a non-linear dimensionality reduction technique that preserves the relationships between data points. It is especially good at visualising complex data, but it can be slow for large data sets and may distort global structures.
- **UMAP (Uniform Manifold Approximation and Projection)**: another non-linear dimensionality reduction method, like t-SNE but more efficient at preserving both local and global structures in the data. UMAP is widely used for visualising high-dimensional data and discovering clusters or patterns.

Embedding: a way to represent words or phrases as vectors (numbers) in a high-dimensional space, allowing language models to understand relationships between words. A smaller Euclidean distance between vectors indicates a closer semantic relationship between words. For example, the word embedding for "football" is closer to "basketball" and further from "monsoon", which is closer to "cloud". These embeddings allow the use of algebraic techniques to convey relationships between words. For example, the embeddings for countries and capitals would obey: $\overrightarrow{Seoul} = \overrightarrow{Korea} - \overrightarrow{Spain} + \overrightarrow{Madrid}$, consistent with the notion that Seoul is related to Korea in the same way that Madrid is related to Spain. Similarly, the linear algebra applies, eg $\overrightarrow{Seoul} - \overrightarrow{Korea} = \overrightarrow{Madrid} - \overrightarrow{Spain}$. Furthermore, embeddings for sentences, paragraphs or any group of words can be computed as a weighted sum of the word embeddings, to represent their collective meaning.

Few-shot learning: a model's ability to learn in-context from only a few examples of a new task, generalising effectively from minimal data.

Fine-tuning: the process of adjusting a pretrained language model for a specific task by training it further on task-specific data.

GPT (generative pretrained transformer): the underlying model of ChatGPT that generates human-like text based on the input it receives, commonly used in text-generation tasks. GPT uses only past words to create the contextualised embedding for each word and, consequently, can be used to generate text.

In-context learning: a model's ability to learn how to solve a task by leveraging examples provided in its input, without requiring additional training or modification of its parameters.

Lemmatisation: a process in NLP that reduces words to their base or root form (eg "running" becomes "run").

Language model (LM): represents a recursive and probabilistic framework of a language. It calculates the probability of the next word based on the preceding words in a text, expressed as $P(w_m | w_{m-1}, w_{m-2}, \dots)$, where w_i denotes the i^{th} word in the sequence.

Large language model (LLM): extends LMs by employing a neural network, more specifically the transformer architecture, to compute these probabilities and uses vast amounts of text to understand and generate human language. See examples of recent LLMs in the table below.

Comparison of selected large language models

Table 1

Model (year)	Pretraining data (# of words)	Max size (# of params)	Context window (# of words)[2]	Max embedding dimensions
BERT (2018)	Bookcorpus, Wikipedia (about 3 bn)	340m	358	1,024
RoBERTa (2019)	BERT + news + websites + stories (about 30bn)	355m	358	1,024
GPT-3 (2019)[1]	Comprehensive (200bn+)	175bn	1,434	12,288
GPT-4 (2023)[1]	Comprehensive (unknown)	1trn+	90,000	3,072[3]
Llama 3.1 (2024)	Comprehensive (10trn+)	405bn	90,000	4,096[3]
Gemini 1.5 (2024)[1]	Comprehensive (unknown)	unknown	700,000	768[3]
Claude 3.5 Sonnet (2024)[1]	Comprehensive (unknown)	unknown	140,000	unknown

[1] The specifics of these proprietary models are not publicly available. [2] Estimate based on an average conversion of 1 token to 0.7 words. [3] Based on the maximum dimensionality offered by their embedding model.

Sources: T Brown et al (2020), "Language models are few-shot learners", arXiv, no 2005.14165 [cs.CL]; J Devlin, M-W Chang, K Lee and K Toutanova (2018), "BERT: pre-training of deep bidirectional transformers for language understanding", arXiv, no 1810.04805 [cs.CL]; Y Liu, Y, M Ott, N Goyal, J Du, M Joshi, D Chen, O Levy, M Lewis, L Zettlemoyer and V Stoyanov (2019), "RoBERTa: a robustly optimized BERT pretraining approach", arXiv, no 1907.11692 [cs.CL]; OpenAI et al (2023), "GPT-4 technical report", arXiv, no 2303.08774 [cs.CL]; Anthropic; Google; Meta.

Named-entity recognition (NER): an NLP task that identifies and classifies key entities (like people, places or organisations) within a text.

Natural language processing (NLP): a field of artificial intelligence focused on the interaction between computers and human language. It involves techniques for processing, analysing and generating text or speech in natural language, enabling tasks such as translation, sentiment analysis, text summarisation and answering questions.

Neural network: see artificial neural network.

Parameters (of an LLM): like coefficients in an econometric model, LLM parameters adjust to learn patterns in language data. More parameters increase the model's capacity to capture nuanced language but also raise training complexity and computational demands.

Supervised learning: a machine learning approach where models are trained on labelled data, meaning they learn from examples with known answers.

Topic modelling: a technique used to identify themes or topics within large collections of text data. Common examples include:

- **latent Dirichlet allocation (LDA)**: a popular topic modelling algorithm that assumes each document is a mixture of several topics, and each topic is a mixture of words. LDA works by identifying patterns of word co-occurrences in a collection of documents to group related words into topics, then estimating the proportion of each topic present in individual documents. The algorithm uses a probabilistic approach, assigning words to topics based on their likelihood of belonging to a topic, and it iteratively refines these assignments to improve the accuracy of the topic distributions.

- **BERTopic**: a modern topic modelling technique that leverages embeddings and clustering algorithms to discover and identify topics in large collections of text. Unlike traditional methods like LDA, which rely on word co-occurrence patterns, BERTopic uses more sophisticated semantic representations of words and sentences to group similar text into meaningful topics. For more details: maartengr.github.io/BERTopic/.

Tokenisation: a process of breaking words down into smaller units called tokens, such as words or sub-words. For example, the word "disinflationary" might be tokenised into "dis-" (prefix), "inflation" (base word) and "-ary" (suffix).

Transformer: a type of neural network architecture that employs multiple interlinked attention and feed-forward neural network layers to generate an embedding for each input word. The transformer model was initially proposed for translation and is not a language model. It comprises an encoder network for the input language and a decoder network for the output language. The decoder network can be directly repurposed as a language model, as it probabilistically predicts the next word based on its context.

Transformer-based language models become particularly useful when they are sufficiently large and have been trained on extensive data sets. The capacity and quality of these models are directly linked to the size and quality of the training data, making them less sensitive to the specificities of the model architecture.

The transformer architecture: mathematical and algorithmic overview

The transformer is a specific neural network structure. Each of its building blocks is a matrix multiplication followed by a non-linearity. We outline the mathematical equations for the transformer building blocks step by step:

1. Provide an embedding for the i-th word: x_i. It is initialised randomly with trainable coefficients. x_i is a row vector of dimension d.
2. Add positional encoding to each word embedding: $x_i \longleftarrow x_i + p_i$.
   p_i is a fixed or trainable vector unique for each position, also a row vector of dimension d.
3. Stack embeddings of n input words into a matrix X, of dimensions $n \times d$.
4. Compute attention for j-th head: $Z_j = \left[Softmax \left({(XW_j^Q)(XW_j^K)^T} \Big/ {\sqrt{d_k}} \right) \right] (XW_j^V)$. The matrices W_j^Q, W_j^K and $W_j^V$are $d \times d_k$ dimensional training parameters. The softmax function, a generalised form of the sigmoid or the logistic function for multiple classes, normalises a vector of values to ensure they lie between 0 and 1 and sum to 1. When applied to a matrix, the softmax operates row-wise, treating each row as a vector. Z_j is of size $n \times d_k$
5. Concatenate the m attention heads: $Z = [Z_1, Z_2, \dots, Z_m\ ]W$. The $md_k \times d$ coefficients of matrix W are trainable parameters. Z is back to the same size as X ($n \times d$).
6. Residual connection: add the attention output Z back to the input. $X \leftarrow X + Z$.
7. Apply a feed-forward neural network independently to each word with residual connection: $x_i \longleftarrow x_i + \max(0, x_i W_1 + b_1)\, W_2 + b_2$, where W_1, W_2, b_1 and b_2 are trainable parameters of size $d \times d_{ff}$, $d_{ff} \times d$, $1 \times d_{ff}$ and $1 \times d$, respectively. The max operation is applied independently element-wise to the d_{ff}-dimensional vector $x_i W_1 + b_1$.
8. Repeat steps 4 to 7 for n_{layers} times, using a different set of parameters for each layer.

In the case of GPT-3 (Brown et al (2020)), the following numbers apply to the structure above:

$d = 12{,}288$, $n = 2{,}048$, $d_k = 128$, $m = 96$, $n_{layers} = 96$, and $d_{ff} = 4d$.

This results in approximately 175 billion parameters.

Unsupervised learning: a machine learning approach where models learn from data that are not labelled, discovering hidden patterns or structures.

Zero-shot learning: the ability of a model to perform tasks it has not explicitly been trained on, without needing examples or prior knowledge.