

Title: BERT (language model)

URL: [https://en.wikipedia.org/wiki/BERT_\(language_model\)](https://en.wikipedia.org/wiki/BERT_(language_model))

PageID: 62026514

Categories: Category:2018 in artificial intelligence, Category:2018 software, Category:Google software, Category:Large language models

Source: Wikipedia (CC BY-SA 4.0).

Bidirectional encoder representations from transformers (BERT) is a language model introduced in October 2018 by researchers at Google . [1] [2] It learns to represent text as a sequence of vectors using self-supervised learning . It uses the encoder-only transformer architecture. BERT dramatically improved the state of the art for large language models . As of 2020 [update] , BERT is a ubiquitous baseline in natural language processing (NLP) experiments. [3]

BERT is trained by masked token prediction and next sentence prediction. With this training, BERT learns contextual, latent representations of tokens in their context, similar to ELMo and GPT-2 . [4] It found applications for many natural language processing tasks, such as coreference resolution and polysemy resolution. [5] It improved on ELMo and spawned the study of "BERTology", which attempts to interpret what is learned by BERT. [3]

BERT was originally implemented in the English language at two model sizes, BERT BASE (110 million parameters) and BERT LARGE (340 million parameters). Both were trained on the Toronto BookCorpus [6] (800M words) and English Wikipedia (2,500M words). [1] : 5 The weights were released on GitHub . [7] On March 11, 2020, 24 smaller models were released, the smallest being BERT TINY with just 4 million parameters. [7]

Architecture

BERT is an "encoder-only" transformer architecture. At a high level, BERT consists of 4 modules:

Tokenizer: This module converts a piece of English text into a sequence of integers ("tokens").

Embedding : This module converts the sequence of tokens into an array of real-valued vectors representing the tokens. It represents the conversion of discrete token types into a lower-dimensional Euclidean space .

Encoder: a stack of Transformer blocks with self-attention , but without causal masking.

Task head: This module converts the final representation vectors into one-hot encoded tokens again by producing a predicted probability distribution over the token types. It can be viewed as a simple decoder, decoding the latent representation into token types, or as an "un-embedding layer".

The task head is necessary for pre-training, but it is often unnecessary for so-called "downstream tasks," such as question answering or sentiment classification . Instead, one removes the task head and replaces it with a newly initialized module suited for the task, and finetune the new module. The latent vector representation of the model is directly fed into this new module, allowing for sample-efficient transfer learning . [1] [8]

Embedding

This section describes the embedding used by BERT BASE . The other one, BERT LARGE , is similar, just larger.

The tokenizer of BERT is WordPiece, which is a sub-word strategy like byte-pair encoding . Its vocabulary size is 30,000, and any token not appearing in its vocabulary is replaced by [UNK] ("unknown").

The first layer is the embedding layer, which contains three components: token type embeddings, position embeddings, and segment type embeddings.

Token type: The token type is a standard embedding layer, translating a one-hot vector into a dense vector based on its token type.

Position: The position embeddings are based on a token's position in the sequence. BERT uses absolute position embeddings, where each position in a sequence is mapped to a real-valued vector. Each dimension of the vector consists of a sinusoidal function that takes the position in the sequence as input.

Segment type: Using a vocabulary of just 0 or 1, this embedding layer produces a dense vector based on whether the token belongs to the first or second text segment in that input. In other words, type-1 tokens are all tokens that appear after the [SEP] special token. All prior tokens are type-0.

The three embedding vectors are added together representing the initial token representation as a function of these three pieces of information. After embedding, the vector representation is normalized using a LayerNorm operation, outputting a 768-dimensional vector for each input token. After this, the representation vectors are passed forward through 12 Transformer encoder blocks, and are decoded back to 30,000-dimensional vocabulary space using a basic affine transformation layer.

Architectural family

The encoder stack of BERT has 2 free parameters: L , the number of layers, and H , the hidden size. There are always $H / 64$ self-attention heads, and the feed-forward/filter size is always $4H$. By varying these two numbers, one obtains an entire family of BERT models. [9]

For BERT:

the feed-forward size and filter size are synonymous. Both of them denote the number of dimensions in the middle layer of the feed-forward network.

the hidden size and embedding size are synonymous. Both of them denote the number of real numbers used to represent a token.

The notation for encoder stack is written as L/H . For example, BERT BASE is written as 12L/768H, BERT LARGE as 24L/1024H, and BERT TINY as 2L/128H.

Training

Pre-training

BERT was pre-trained simultaneously on two tasks: [10]

Masked language modeling (MLM): In this task, BERT ingests a sequence of words, where one word may be randomly changed ("masked"), and BERT tries to predict the original words that had been changed. For example, in the sentence "The cat sat on the [MASK] ," BERT would need to predict "mat." This helps BERT learn bidirectional context, meaning it understands the relationships between words not just from left to right or right to left but from both directions at the same time.

Next sentence prediction (NSP): In this task, BERT is trained to predict whether one sentence logically follows another. For example, given two sentences, "The cat sat on the mat" and "It was a sunny day", BERT has to decide if the second sentence is a valid continuation of the first one. This helps BERT understand relationships between sentences, which is important for tasks like question answering or document classification.

Masked language modeling

In masked language modeling, 15% of tokens would be randomly selected for masked-prediction task, and the training objective was to predict the masked token given its context. In more detail, the selected token is:

replaced with a [MASK] token with probability 80%,

replaced with a random word token with probability 10%,

not replaced with probability 10%.

The reason not all selected tokens are masked is to avoid the dataset shift problem. The dataset shift problem arises when the distribution of inputs seen during training differs significantly from the distribution encountered during inference. A trained BERT model might be applied to word representation (like Word2Vec), where it would be run over sentences not containing any [MASK] tokens. It is later found that more diverse training objectives are generally better. [11]

As an illustrative example, consider the sentence "my dog is cute". It would first be divided into tokens like "my 1 dog 2 is 3 cute 4 ". Then a random token in the sentence would be picked. Let it be the 4th one "cute 4 ". Next, there would be three possibilities:

with probability 80%, the chosen token is masked, resulting in "my 1 dog 2 is 3 [MASK] 4 ";

with probability 10%, the chosen token is replaced by a uniformly sampled random token, such as "happy", resulting in "my 1 dog 2 is 3 happy 4 ";

with probability 10%, nothing is done, resulting in "my 1 dog 2 is 3 cute 4 ".

After processing the input text, the model's 4th output vector is passed to its decoder layer, which outputs a probability distribution over its 30,000-dimensional vocabulary space.

Next sentence prediction

Given two sentences, the model predicts if they appear sequentially in the training corpus, outputting either [IsNext] or [NotNext]. During training, the algorithm sometimes samples two sentences from a single continuous span in the training corpus, while at other times, it samples two sentences from two discontinuous spans.

The first sentence starts with a special token, [CLS] (for "classify"). The two sentences are separated by another special token, [SEP] (for "separate"). After processing the two sentences, the final vector for the [CLS] token is passed to a linear layer for binary classification into [IsNext] and [NotNext].

For example:

Given "[CLS] my dog is cute [SEP] he likes playing [SEP] ", the model should predict [IsNext].

Given "[CLS] my dog is cute [SEP] how do magnets work [SEP] ", the model should predict [NotNext].

Fine-tuning

Sentiment classification

Sentence classification

Answering multiple-choice questions

Part-of-speech tagging

BERT is meant as a general pretrained model for various applications in natural language processing. That is, after pre-training, BERT can be fine-tuned with fewer resources on smaller datasets to optimize its performance on specific tasks such as natural language inference and text classification, and sequence-to-sequence-based language generation tasks such as question answering and conversational response generation. [12]

The original BERT paper published results demonstrating that a small amount of finetuning (for BERT LARGE, 1 hour on 1 Cloud TPU) allowed it to achieve state-of-the-art performance on a number of natural language understanding tasks: [1]

GLUE (General Language Understanding Evaluation) task set (consisting of 9 tasks);

SQuAD (Stanford Question Answering Dataset [13]) v1.1 and v2.0;

SWAG (Situations With Adversarial Generations [14]).

In the original paper, all parameters of BERT are fine-tuned, and recommended that, for downstream applications that are text classifications, the output token at the [CLS] input token is fed into a linear-softmax layer to produce the label outputs. [1]

The original code base defined the final linear layer as a "pooler layer", in analogy with global pooling in computer vision, even though it simply discards all output tokens except the one corresponding to [CLS] . [15]

Cost

BERT was trained on the BookCorpus (800M words) and a filtered version of English Wikipedia (2,500M words) without lists, tables, and headers.

Training BERT BASE on 4 cloud TPU (16 TPU chips total) took 4 days, at an estimated cost of 500 USD. [7] Training BERT LARGE on 16 cloud TPU (64 TPU chips total) took 4 days. [1]

Interpretation

Language models like ELMo, GPT-2, and BERT, spawned the study of "BERTology", which attempts to interpret what is learned by these models. Their performance on these natural language understanding tasks are not yet well understood. [3] [16] [17] Several research publications in 2018 and 2019 focused on investigating the relationship behind BERT's output as a result of carefully chosen input sequences, [18] [19] analysis of internal vector representations through probing classifiers, [20] [21] and the relationships represented by attention weights. [16] [17]

The high performance of the BERT model could also be attributed to the fact that it is bidirectionally trained. [22] This means that BERT, based on the Transformer model architecture, applies its self-attention mechanism to learn information from a text from the left and right side during training, and consequently gains a deep understanding of the context. For example, the word fine can have two different meanings depending on the context (I feel fine today , She has fine blond hair). BERT considers the words surrounding the target word fine from the left and right side.

However it comes at a cost: due to encoder-only architecture lacking a decoder, BERT can't be prompted and can't generate text , while bidirectional models in general do not work effectively without the right side, thus being difficult to prompt. As an illustrative example, if one wishes to use BERT to continue a sentence fragment "Today, I went to", then naively one would mask out all the tokens as "Today, I went to [MASK] [MASK] [MASK] ... [MASK] ." where the number of [MASK] is the length of the sentence one wishes to extend to. However, this constitutes a dataset shift, as during training, BERT has never seen sentences with that many tokens masked out. Consequently, its performance degrades. More sophisticated techniques allow text generation, but at a high computational cost. [23]

History

BERT was originally published by Google researchers Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. The design has its origins from pre-training contextual representations, including semi-supervised sequence learning , [24] generative pre-training, ELMo , [25] and ULMFit. [26] Unlike previous models, BERT is a deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus . Context-free models such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary, whereas BERT takes into account the context for each occurrence of a given word. For instance, whereas the vector for "running" will have the same word2vec vector representation for both of its occurrences in the sentences "He is running a company" and "He is running a marathon", BERT will provide a contextualized embedding that will be different according to the sentence. [4]

On October 25, 2019, Google announced that they had started applying BERT models to English-language search queries on Google Search within the US. [27] On December 9, 2019, it was reported that BERT had been adopted by Google Search for over 70 languages. [28] [29] In October 2020, almost every single English-based query was processed by a BERT model. [30]

Variants

The BERT models were influential and inspired many variants.

RoBERTa (2019) [31] was an engineering improvement. It preserves BERT's architecture (slightly larger, at 355M parameters), but improves its training, changing key hyperparameters, removing the next-sentence prediction task, and using much larger mini-batch sizes.

XLNet (2019) [32] was a multilingual RoBERTa model. It was one of the first works on multilingual language modeling at scale.

DistilBERT (2019) distills BERT BASE to a model with just 60% of its parameters (66M), while preserving 95% of its benchmark scores. [33] [34] Similarly, TinyBERT (2019) [35] is a distilled model with just 28% of its parameters.

ALBERT (2019) [36] used shared-parameter across layers, and experimented with independently varying the hidden size and the word-embedding layer's output size as two hyperparameters. They also replaced the next sentence prediction task with the sentence-order prediction (SOP) task, where the model must distinguish the correct order of two consecutive text segments from their reversed order.

ELECTRA (2020) [37] applied the idea of generative adversarial networks to the MLM task. Instead of masking out tokens, a small language model generates random plausible substitutions, and a larger network identify these replaced tokens. The small model aims to fool the large model.

DeBERTa (2020) [38] is a significant architectural variant, with disentangled attention . Its key idea is to treat the positional and token encodings separately throughout the attention mechanism. Instead of combining the positional encoding (x_{position}) and token encoding (x_{token}) into a single input vector ($x_{\mathrm{input}} = x_{\mathrm{position}} + x_{\mathrm{token}}$), DeBERTa keeps them separate as a tuple: ($x_{\mathrm{position}}, x_{\mathrm{token}}$) . Then, at each self-attention layer, DeBERTa computes three distinct attention matrices, rather than the single attention matrix used in BERT: [note 1]

The three attention matrices are added together element-wise, then passed through a softmax layer and multiplied by a projection matrix.

Absolute position encoding is included in the final self-attention layer as additional input.

Notes

References

Further reading

Rogers, Anna; Kovaleva, Olga; Rumshisky, Anna (2020). "A Primer in BERTology: What we know about how BERT works". arXiv : 2002.12327 [cs.CL].

External links

Official GitHub repository

v

t

e

Google

Google Brain

Google DeepMind

AlphaGo (2015)

Master (2016)

AlphaGo Zero (2017)

AlphaZero (2017)

MuZero (2019)

Fan Hui (2015)

Lee Sedol (2016)
Ke Jie (2017)
AlphaGo (2017)
The MANIAC (2023)
AlphaFold (2018)
AlphaStar (2019)
AlphaDev (2023)
AlphaGeometry (2024)
AlphaGenome (2025)
Inception (2014)
WaveNet (2016)
MobileNet (2017)
Transformer (2017)
EfficientNet (2019)
Gato (2022)
Quantum Artificial Intelligence Lab
TensorFlow
Tensor Processing Unit
Assistant (2016)
Sparrow (2022)
Gemini (2023)
BERT (2018)
XLNet (2019)
T5 (2019)
LaMDA (2021)
Chinchilla (2022)
PaLM (2022)
Imagen (2023)
Gemini (2023)
VideoPoet (2024)
Gemma (2024)
Veo (2024)
DreamBooth (2022)
NotebookLM (2023)
Vids (2024)
Gemini Robotics (2025)
" Attention Is All You Need "
Future of Go Summit

Generative pre-trained transformer

Google Labs

Google Pixel

Google Workspace

Robot Constitution

Category

Commons

v

t

e

AI

Area 120

ATAP

Brain

China

Cloud Platform

Energy

Google.org Crisis Response

Crisis Response

Health

Registry

DeepMind

Fitbit

ITA Software

Jigsaw

Looker

Mandiant

Security Operations

Owlchemy Labs

Actifio

Adscape

Akwan Information Technologies

Anvato

Apigee

BandPage

Bitium

BufferBox

Crashlytics

Dodgeball
DoubleClick
Dropcam
Endoxon
Flutter
Global IP Solutions
Green Throttle Games
GreenBorder
Gridcentric
ImageAmerica
Impermium
Invite Media
Kaltix
Marratech
Meebo
Metaweb
Neotonic Software
Neverware
Nik Software
Orbitera
Pyra Labs
Quest Visual
Reqwireless
RightsFlow
Sidewalk Labs
SlickLogin
Titan Aerospace
Typhoon Studios
Urban Engines
Vicarious
Viewdle
Wavii
Wildfire Interactive
YouTube Next Lab and Audience Development Group
Business Groups
Computing University Initiative
Contact Lens
Content ID

CrossCheck
Data Liberation Front
Data Transfer Project
Developer Expert
DigiKavach
DigiPivot
Digital Garage
Digital News Initiative
Digital Unlocked
Dragonfly
Founders' Award
Free Zone
Get Your Business Online
Google for Education
Google for Startups
Living Stories
Made with Code
News Lab
PowerMeter
Privacy Sandbox
Project Nightingale
Project Nimbus
Project Sunroof
Project Zero
Quantum Artificial Intelligence Lab
RechargeIT
Sensorvault
Silicon Initiative
Solve for X
Street View Trusted
Student Ambassador Program
Vevo
YouTube BrandConnect
YouTube Creator Awards
YouTube Select
YouTube Original Channel Initiative
Year in Search
YouTube Rewind 2018 2019

2018

2019

AlphaGo versus Fan Hui

AlphaGo versus Lee Sedol

AlphaGo versus Ke Jie

Android Developer Challenge

Android Developer Day

Android Developer Lab

CNN/YouTube presidential debates

Code-in

Code Jam

Developer Day

Developers Live

Doodle4Google

Future of Go Summit

G-Day

Hash Code

I/O

Lunar X Prize

Mapathon

Science Fair

Summer of Code

World Chess Championship 2024

YouTube Awards

YouTube Comedy Week

YouTube Live

YouTube Music Awards 2013 2015

2013

2015

YouTube Space Lab

YouTube Symphony Orchestra

111 Eighth Avenue

Android lawn statues

Androidland

Barges

Binoculars Building

Central Saint Giles

Chelsea Market

Chrome Zone
Data centers
GeoEye-1
Googleplex
Ivanpah Solar Power Facility
James R. Thompson Center
King's Cross
Mayfield Mall
Pier 57
Sidewalk Toronto
St. John's Terminal
Submarine cables Dunant Grace Hopper Unity
Dunant
Grace Hopper
Unity
WiFi
YouTube Space
YouTube Theater
Krishna Bharat
Vint Cerf
Jeff Dean
John Doerr
Sanjay Ghemawat
Al Gore
John L. Hennessy
Urs Hölzle
Salar Kamangar
Ray Kurzweil
Ann Mather
Alan Mulally
Rick Osterloh
Sundar Pichai (CEO)
Ruth Porat (CFO)
Rajen Sheth
Hal Varian
Neal Mohan
Andy Bechtolsheim
Sergey Brin (co-founder)

David Cheriton
Matt Cutts
David Drummond
Alan Eustace
Timnit Gebru
Omid Kordestani
Paul Otellini
Larry Page (co-founder)
Patrick Pichette
Eric Schmidt
Ram Shriram
Amit Singhal
Shirley M. Tilghman
Rachel Whetstone
Susan Wojcicki
Censorship
DeGoogle
FairSearch
" Google's Ideological Echo Chamber "
No Tech for Apartheid
Privacy concerns Street View YouTube
Street View
YouTube
Trade unions Alphabet Workers Union
Alphabet Workers Union
YouTube copyright issues
Backdoor advertisement controversy
Blocking of YouTube videos in Germany
Data breach
Elsagate
Fantastic Adventures scandal
Kohistan video case
Reactions to Innocence of Muslims
San Francisco tech bus protests
Services outages
Slovenian government incident
Walkouts
YouTube headquarters shooting

Android apps
April Fools' Day jokes
Doodles Doodle Champion Island Games Magic Cat Academy
Doodle Champion Island Games
Magic Cat Academy
Easter eggs
History Gmail Search YouTube
Gmail
Search
YouTube
Logo
Material Design
Mergers and acquisitions
Accelerated Linear Algebra
AMP
Actions on Google
ALTS
American Fuzzy Lop
Android Cloud to Device Messaging
Android Debug Bridge
Android NDK
Android Runtime
Android SDK
Android Studio
Angular
AngularJS
Apache Beam
APIs
App Engine
App Inventor
App Maker
App Runtime for Chrome
AppJet
Apps Script
AppSheet
ARCore
Base
Bazel

BeyondCorp
Bigtable
BigQuery
Bionic
Blockly
Borg
Caja
Cameyo
Chart API
Charts
Chrome Frame
Chromium Blink
Blink
Closure Tools
Cloud Connect
Cloud Dataflow
Cloud Datastore
Cloud Messaging
Cloud Shell
Cloud Storage
Code Search
Compute Engine
Cpplint
Dalvik
Data Protocol
Dialogflow
Exposure Notification
Fast Pair
Fastboot
Federated Learning of Cohorts
File System
Firebase
Firebase Studio
Firebase Cloud Messaging
FlatBuffers
Flutter
Freebase
Gadgets

Ganeti
Gears
Gerrit
GLOP
gRPC
Gson
Guava
Guetzli
Guice
gVisor
GYP
JAX
Jetpack Compose
Keyhole Markup Language
Kubernetes
Kythe
LevelDB
Lighthouse
Looker Studio
Imctfy
MapReduce
Mashup Editor
Matter
Mobile Services
Namebench
Native Client
Neatx
Neural Machine Translation
Nomulus
Open Location Code
OpenRefine
OpenSocial
Optimize
OR-Tools
Pack
PageSpeed
Piper
Plugin for Eclipse

Polymer
Programmable Search Engine
Project Shield
Public DNS
reCAPTCHA
RenderScript
SafetyNet
SageTV
Schema.org
Search Console
Shell
Sitemaps
Skia Graphics Engine
Spanner
Sputnik
Stackdriver
Swifty
Tango
TensorFlow
Tesseract
Test
Translator Toolkit
Urchin UTM parameters
UTM parameters
V8
VirusTotal
VisBug
Wave Federation Protocol
Weave
Web Accelerator
Web Designer
Web Server
Web Toolkit
Webdriver Torso
WebRTC
Android Cupcake Donut Eclair Froyo Gingerbread Honeycomb Ice Cream Sandwich Jelly Bean
KitKat Lollipop Marshmallow Nougat Oreo Pie 10 11 12 13 14 15 16 version history smartphones
Cupcake

Donut
Eclair
Froyo
Gingerbread
Honeycomb
Ice Cream Sandwich
Jelly Bean
KitKat
Lollipop
Marshmallow
Nougat
Oreo
Pie
10
11
12
13
14
15
16
version history
smartphones
Android Automotive
Android Go devices
devices
Android Things
Android TV devices
devices
Android XR
ChromeOS
ChromeOS Flex
ChromiumOS
Fuchsia
Glass OS
gLinux
Goobuntu
TV
Wear OS

BERT
Chinchilla
DreamBooth
Gemini
Gemma
Imagen (2023)
LaMDA
PaLM
T5
Veo (text-to-video model)
VideoPoet
XLNet
EfficientNet
Gato
Inception
MobileNet
Transformer
WaveNet
AlphaDev
AlphaFold
AlphaGeometry
AlphaGo
AlphaGo Zero
AlphaStar
AlphaZero
Master
MuZero
AAB
APK
AV1
iLBC
iSAC
libvpx
Lyra
Protocol Buffers
Ultra HDR
VP3
VP6

VP8
VP9
WebM
WebP
WOFF2
Carbon
Dart
Go
Sawzall
Googlebot
Hummingbird
Mobilegeddon
PageRank matrix
matrix
Panda
Penguin
Pigeon
RankBrain
.app
.dev
.google
.zip
g.co
google.by
Croscore
Noto
Product Sans
Roboto
Aardvark
Account Dashboard Takeout
Dashboard
Takeout
Ad Manager
AdMob
Ads
AdSense
Affiliate Network
Alerts

Allo
Analytics
Android Auto
Android Beam
Answers
Apture
Arts & Culture
Assistant
Attribution
Authenticator
BebaPay
BeatThatQuote.com
Beam
Blog Search
Blogger
Body
Bookmarks
Books Ngram Viewer
Ngram Viewer
Browser Sync
Building Maker
Bump
BumpTop
Buzz
Calendar
Cast
Catalogs
Chat
Checkout
Chrome
Chrome Apps
Chrome Experiments
Chrome Remote Desktop
Chrome Web Store
Classroom
Cloud Print
Cloud Search
Contacts

Contributor
Crowdsource
Currents (social app)
Currents (news app)
Data Commons
Dataset Search
Desktop
Dictionary
Dinosaur Game
Directory
Docs
Docs Editors
Domains
Drawings
Drive
Duo
Earth
Etherpad
Expeditions
Express
Family Link
Fast Flip
FeedBurner
fflick
Fi Wireless
Finance
Files
Find Hub
Fit
Flights
Flu Trends
Fonts
Forms
Friend Connect
Fusion Tables
Gboard
Gemini
Gesture Search

Gizmo5
Google+
Gmail
Goggles
GOOG-411
Grasshopper
Groups
Hangouts
Helpouts
iGoogle
Images Image Labeler
Image Labeler
Image Swirl
Inbox by Gmail
Input Tools Japanese Input Pinyin
Japanese Input
Pinyin
Insights for Search
Jaiku
Jamboard
Kaggle
Keep
Knol
Labs
Latitude
Lens
Like.com
Live Transcribe
Lively
Map Maker
Maps
Maps Navigation
Marketing Platform
Meet
Messages
Moderator
My Tracks
Nearby Share

News
News & Weather
News Archive
Notebook
NotebookLM
Now
Offers
One
One Pass
Opinion Rewards
Orkut
Oyster
Panoramio
PaperofRecord.com
Patents
Page Creator
Pay (mobile app)
Pay (payment method)
Pay Send
People Cards
Person Finder
Personalized Search
Photomath
Photos
Picasa
Picasa Web Albums
Picnik
Pixel Camera
Play
Play Books
Play Games
Play Music
Play Newsstand
Play Pass
Play Services
Podcasts
Poly
Postini

PostRank
Primer
Public Alerts
Public Data Explorer
Question Hub
Quick, Draw!
Quick Search Box
Quick Share
Quickoffice
Read Along
Reader
Reply
Safe Browsing
SageTV
Santa Tracker
Schemer
Scholar
Search AI Overviews Knowledge Graph SafeSearch
AI Overviews
Knowledge Graph
SafeSearch
Searchwiki
Sheets
Shoploop
Shopping
Sidewiki
Sites
Slides
Snapseed
Socratic
Softcard
Songza
Sound Amplifier
Spaces
Sparrow (chatbot)
Sparrow (email client)
Speech Recognition & Synthesis
Squared

Stadia
Station
Store
Street View
Surveys
Sync
Tables
Talk
TalkBack
Tasks
Tenor
Tez
Tilt Brush
Toolbar
Toontastic 3D
Translate
Travel
Trendalyzer
Trends
TV
URL Shortener
Video
Vids
Voice
Voice Access
Voice Search
Wallet
Wave
Waze
WDYL
Web Light
Where Is My Train
Widevine
Wiz
Word Lens
Workspace
Workspace Marketplace
YouTube

YouTube Kids
YouTube Music
YouTube Premium
YouTube Shorts
YouTube Studio
YouTube TV
YouTube VR
Pixel (2016)
Pixel 2 (2017)
Pixel 3 (2018)
Pixel 3a (2019)
Pixel 4 (2019)
Pixel 4a (2020)
Pixel 5 (2020)
Pixel 5a (2021)
Pixel 6 (2021)
Pixel 6a (2022)
Pixel 7 (2022)
Pixel 7a (2023)
Pixel Fold (2023)
Pixel 8 (2023)
Pixel 8a (2024)
Pixel 9 (2024)
Pixel 9 Pro Fold (2024)
Pixel 9a (2025)
Pixel 10 (2025)
Pixel 10 Pro Fold (2025)
Pixel Watch (2022)
Pixel Watch 2 (2023)
Pixel Watch 3 (2024)
Pixel Watch 4 (2025)
Pixel C (2015)
Pixel Slate (2018)
Pixel Tablet (2023)
Chromebook Pixel (2013–2015)
Pixelbook (2017)
Pixelbook Go (2019)
Pixel Buds (2017–present)

Nexus One (2010)
Nexus S (2010)
Galaxy Nexus (2011)
Nexus 4 (2012)
Nexus 5 (2013)
Nexus 6 (2014)
Nexus 5X (2015)
Nexus 6P (2015)
Nexus 7 (2012)
Nexus 10 (2012)
Nexus 7 (2013)
Nexus 9 (2014)
Nexus Q (2012)
Nexus Player (2014)
Android Dev Phone
Android One
Cardboard
Chromebit
Chromebook
Chromebox
Chromecast
Clips
Daydream
Fitbit
Glass
Liftware
Liquid Galaxy
Nest smart speakers Thermostat Wifi
smart speakers
Thermostat
Wifi
Play Edition
Project Ara
OnHub
Pixel Visual Core
Project Iris
Search Appliance
Sycamore processor

Tensor

Tensor Processing Unit

Titan Security Key

v

t

e

Feldman v. Google, Inc. (2007)

Rescuecom Corp. v. Google Inc. (2009)

Goddard v. Google, Inc. (2009)

Rosetta Stone Ltd. v. Google, Inc. (2012)

Google, Inc. v. American Blind & Wallpaper Factory, Inc. (2017)

Jedi Blue

European Union (2010–present)

United States v. Adobe Systems, Inc., Apple Inc., Google Inc., Intel Corporation, Intuit, Inc., and Pixar (2011)

Umar Javeed, Sukarma Thapar, Aaqib Javeed vs. Google LLC and Ors. (2019)

United States v. Google LLC (2020)

United States v. Google LLC (2023)

Perfect 10, Inc. v. Amazon.com, Inc. (2007)

Viacom International, Inc. v. YouTube, Inc. (2010)

Lenz v. Universal Music Corp. (2015)

Authors Guild, Inc. v. Google, Inc. (2015)

Field v. Google, Inc. (2016)

Google LLC v. Oracle America, Inc. (2021)

Smartphone patent wars

Rocky Mountain Bank v. Google, Inc. (2009)

Hibnick v. Google, Inc. (2010)

United States v. Google Inc. (2012)

Judgement of the German Federal Court of Justice on Google's autocomplete function (2013)

Joffe v. Google, Inc. (2013)

Mosley v SARL Google (2013)

Google Spain v AEPD and Mario Costeja González (2014)

Frank v. Gaos (2019)

Garcia v. Google, Inc. (2015)

Google LLC v Defteros (2020)

Epic Games v. Google (2021)

Gonzalez v. Google LLC (2022)

Beauty YouTuber

BookTube
BreadTube
" Don't be evil "
Gayglers
Google as a verb
Google bombing 2004 U.S. presidential election
2004 U.S. presidential election
Google effect
Googlefight
Google hacking
Googleshare
Google tax
Googلهhack
Googlization
Illegal flower tribute
Objectives and key results
Rooting
Search engine manipulation effect
Side project time
Sitelink
Site reliability engineering
StudyTube
VTuber
YouTube Poop
YouTuber list
list
Bootling process
Custom distributions
Features
Recovery mode
Software development
Africa
Antarctica
Asia Israel
Israel
Europe
North America Canada United States
Canada

United States

Oceania

South America Argentina Chile Colombia

Argentina

Chile

Colombia

Copyright strike

Education

Features

Moderation

Most-disliked videos

Most-liked videos

Most-subscribed channels

Most-viewed channels

Most-viewed videos Arabic music videos Chinese music videos French music videos Indian videos Pakistani videos

Arabic music videos

Chinese music videos

French music videos

Indian videos

Pakistani videos

Official channel

Social impact

YouTube Premium original programming

Gmail interface

Maps pin

Most downloaded Google Play applications

Stadia games

AlphaGo

Google: Behind the Screen

Google Maps Road Trip

Google and the World Brain

The Creepy Line

Google Hacks

The Google Story

Googled: The End of the World as We Know It

How Google Works

I'm Feeling Lucky

In the Plex
The MANIAC
Google Feud
Google Me (film)
" Google Me " (Kim Zolciak song)
" Google Me " (Teyana Taylor song)
Is Google Making Us Stupid?
Proceratium google
Matt Nathanson: Live at Google
The Billion Dollar Code
The Internship
Where on Google Earth is Carmen Sandiego?
" Attention Is All You Need "
elgooG
Generative pre-trained transformer
" Me at the zoo "
Predictions of the end
Relationship with Wikipedia
" Reunion "
Robot Constitution
Category
Outline
v
t
e
AI-complete
Bag-of-words
n -gram Bigram Trigram
Bigram
Trigram
Computational linguistics
Natural language understanding
Stop words
Text processing
Argument mining
Collocation extraction
Concept mining
Coreference resolution

Deep linguistic processing
Distant reading
Information extraction
Named-entity recognition
Ontology learning
Parsing Semantic parsing Syntactic parsing
Semantic parsing
Syntactic parsing
Part-of-speech tagging
Semantic analysis
Semantic role labeling
Semantic decomposition
Semantic similarity
Sentiment analysis
Terminology extraction
Text mining
Textual entailment
Truecasing
Word-sense disambiguation
Word-sense induction
Compound-term processing
Lemmatisation
Lexical analysis
Text chunking
Stemming
Sentence segmentation
Word segmentation
Multi-document summarization
Sentence extraction
Text simplification
Computer-assisted
Example-based
Rule-based
Statistical
Transfer-based
Neural
BERT
Document-term matrix

Explicit semantic analysis
fastText
GloVe
Language model (large)
Latent semantic analysis
Seq2seq
Word embedding
Word2vec
Corpus linguistics
Lexical resource
Linguistic Linked Open Data
Machine-readable dictionary
Parallel text
PropBank
Semantic network
Simple Knowledge Organization System
Speech corpus
Text corpus
Thesaurus (information retrieval)
Treebank
Universal Dependencies
BabelNet
Bank of English
DBpedia
FrameNet
Google Ngram Viewer
UBY
WordNet
Wikidata
Speech recognition
Speech segmentation
Speech synthesis
Natural language generation
Optical character recognition
Document classification
Latent Dirichlet allocation
Pachinko allocation
Automated essay scoring

Concordancer
Grammar checker
Predictive text
Pronunciation assessment
Spell checker
Chatbot
Interactive fiction
Question answering
Virtual assistant
Voice user interface
Formal semantics
Hallucination
Natural Language Toolkit
spaCy
v
t
e
History timeline
timeline
Companies
Projects
Parameter Hyperparameter
Hyperparameter
Loss functions
Regression Bias–variance tradeoff Double descent Overfitting
Bias–variance tradeoff
Double descent
Overfitting
Clustering
Gradient descent SGD Quasi-Newton method Conjugate gradient method
SGD
Quasi-Newton method
Conjugate gradient method
Backpropagation
Attention
Convolution
Normalization Batchnorm
Batchnorm

Activation Softmax Sigmoid Rectifier
Softmax
Sigmoid
Rectifier
Gating
Weight initialization
Regularization
Datasets Augmentation
Augmentation
Prompt engineering
Reinforcement learning Q-learning SARSA Imitation Policy gradient
Q-learning
SARSA
Imitation
Policy gradient
Diffusion
Latent diffusion model
Autoregression
Adversary
RAG
Uncanny valley
RLHF
Self-supervised learning
Reflection
Recursive self-improvement
Hallucination
Word embedding
Vibe coding
Machine learning In-context learning
In-context learning
Artificial neural network Deep learning
Deep learning
Language model Large language model NMT
Large language model
NMT
Reasoning language model
Model Context Protocol
Intelligent agent

Artificial human companion

Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec
Seq2seq
GloVe
BERT
T5
Llama
Chinchilla AI
PaLM
GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5
1
2
3
J
ChatGPT
4
4o
o1
o3
4.5
4.1
o4-mini
5
Claude
Gemini Gemini (language model) Gemma
Gemini (language model)
Gemma
Grok
LaMDA
BLOOM
DBRX
Project Debater
IBM Watson
IBM Watsonx
Granite
PanGu- Σ
DeepSeek
Qwen
AlphaGo

AlphaZero
OpenAI Five
Self-driving car
MuZero
Action selection AutoGPT
AutoGPT
Robot control
Alan Turing
Warren Sturgis McCulloch
Walter Pitts
John von Neumann
Claude Shannon
Shun'ichi Amari
Kunihiko Fukushima
Takeo Kanade
Marvin Minsky
John McCarthy
Nathaniel Rochester
Allen Newell
Cliff Shaw
Herbert A. Simon
Oliver Selfridge
Frank Rosenblatt
Bernard Widrow
Joseph Weizenbaum
Seymour Papert
Seppo Linnainmaa
Paul Werbos
Geoffrey Hinton
John Hopfield
Jürgen Schmidhuber
Yann LeCun
Yoshua Bengio
Lotfi A. Zadeh
Stephen Grossberg
Alex Graves
James Goodnight
Andrew Ng

Fei-Fei Li
Alex Krizhevsky
Ilya Sutskever
Oriol Vinyals
Quoc V. Le
Ian Goodfellow
Demis Hassabis
David Silver
Andrej Karpathy
Ashish Vaswani
Noam Shazeer
Aidan Gomez
John Schulman
Mustafa Suleyman
Jan Leike
Daniel Kokotajlo
François Chollet
Neural Turing machine
Differentiable neural computer
Transformer Vision transformer (ViT)
Vision transformer (ViT)
Recurrent neural network (RNN)
Long short-term memory (LSTM)
Gated recurrent unit (GRU)
Echo state network
Multilayer perceptron (MLP)
Convolutional neural network (CNN)
Residual neural network (RNN)
Highway network
Mamba
Autoencoder
Variational autoencoder (VAE)
Generative adversarial network (GAN)
Graph neural network (GNN)
Category