LLM

large language models
part 1

Plan

- Introduction to LLM
- Key models and architecture evolution:
  - BERT
  - GPT
  - T5
  - BLOOM
  - PaLM
  - FLAN-T5

Introduction to LLM

Definition of LLM

LLM (Large Language Models) are large-scale language models

- Trained on massive amounts of text data
- Contain a large number of model parameters
- Capable of generating, understanding, and processing human language

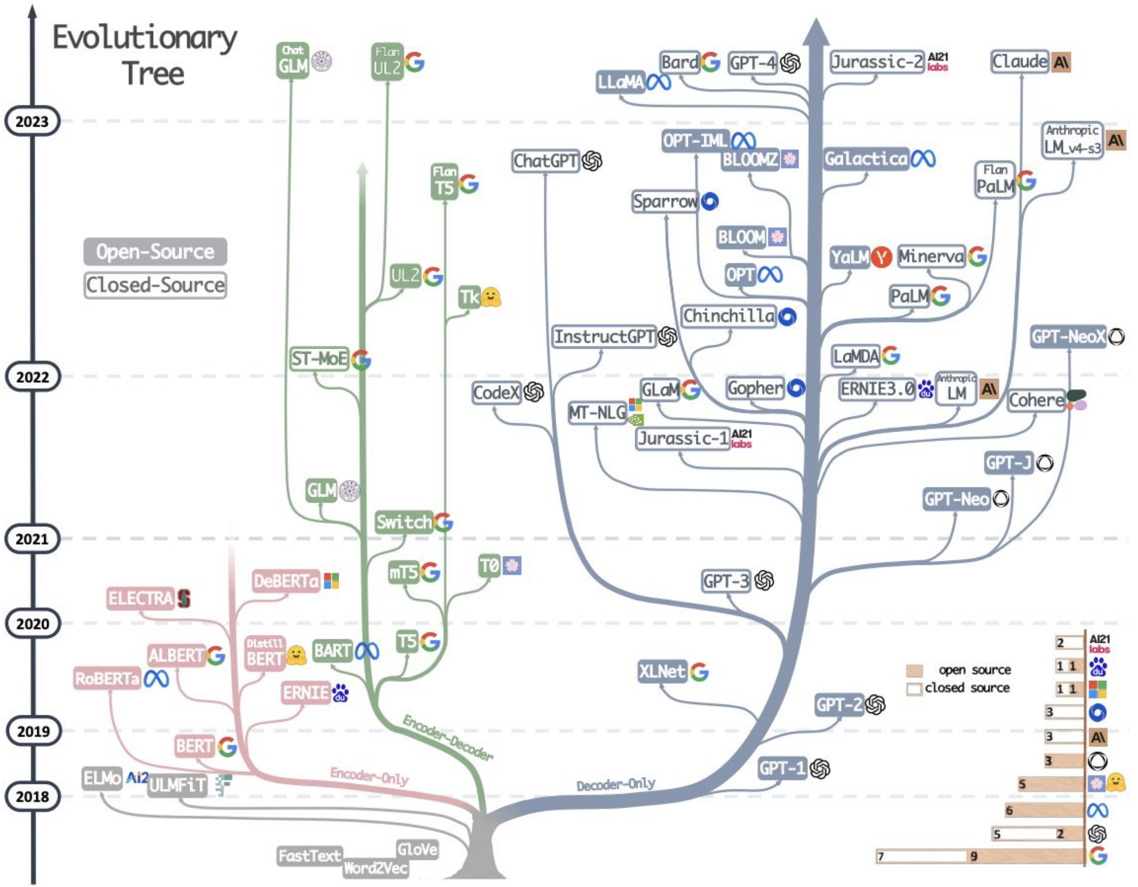
But how did we arrive at this point?

History of Language Model Development

- Decoder-only models (blue branch)
- Encoder-only models (pink branch)
- Encoder-decoder models (green branch)

- Vertical axis shows release dates

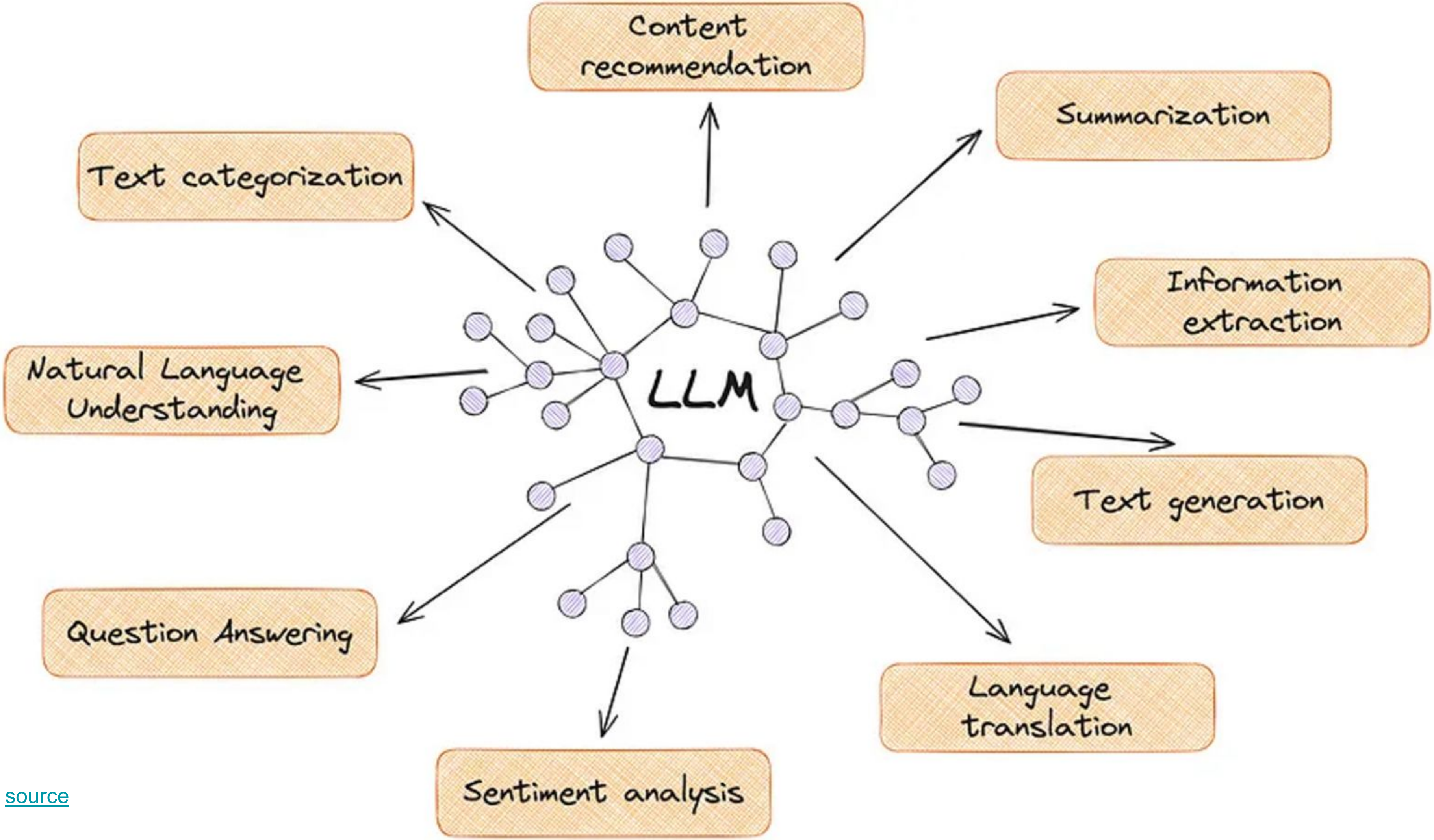
- Filled squares: open-source
- Empty squares: closed-source



source

Applications of LLM in Various Domains

- Machine translation
- Text generation
- Sentiment analysis
- Question answering
- Text summarization
- Named entity recognition
- Chatbots and virtual assistants
- Code writing and debugging

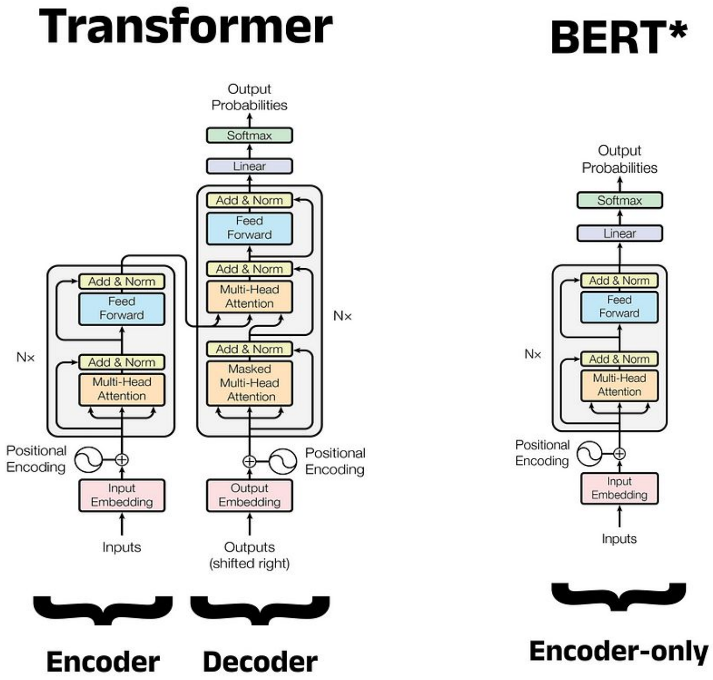

Content recommendation
Summarization
Text categorization
Information extraction
Natural Language Understanding
LLM
Text generation
Question Answering
Language translation
Sentiment analysis

source

Architecture Evolution of Key Models

BERT

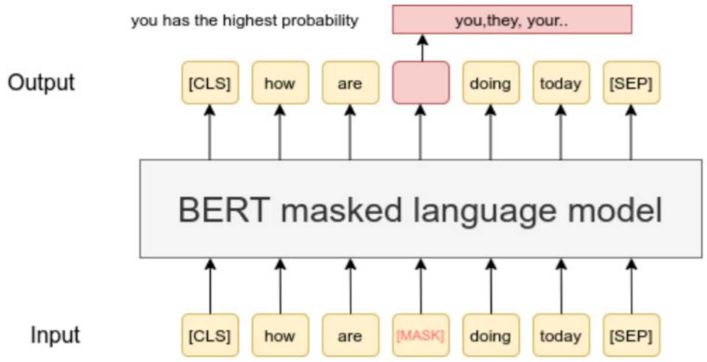
- Encoder-only
- Bidirectional Encoder Representations from Transformers (MLM task)
- Based on Transformer architecture
- Bidirectional context encoding
- Pretrained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks

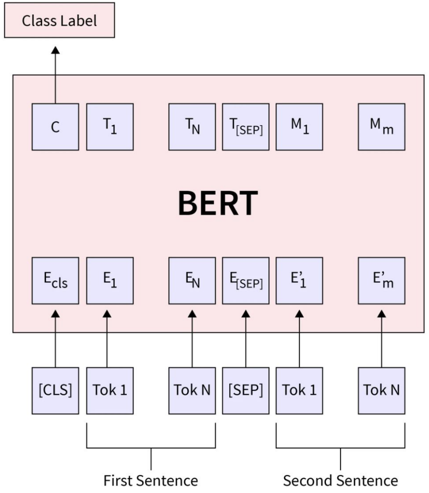


2018

BERT. Pretrain Tasks.

- Masked Language Model (MLM)
  - 15% of tokens are randomly masked
  - 80% replaced with [MASK]
  - 10% replaced with a random word
  - 10% remain unchanged
- Next Sentence Prediction (NSP)
  - Model learns to predict if sentence B logically follows sentence A





source1, source2

BERT. Architecture and Training.

Architecture:

- 12 layers (BERT-base) or 24 layers (BERT-large)
- 768 hidden neurons (base) or 1024 (large)
- 12 attention heads (base) or 16 (large)

Training Details:

- Batch size: 256 sequences
- 1,000,000 training steps
- Optimizer: Adam with β1 = 0.9, β2 = 0.999
- Learning rate: 1e-4
- Dropout: 0.1 across all layers
- Activation: GELU

BERT. Corpora and Data.

Input Data:

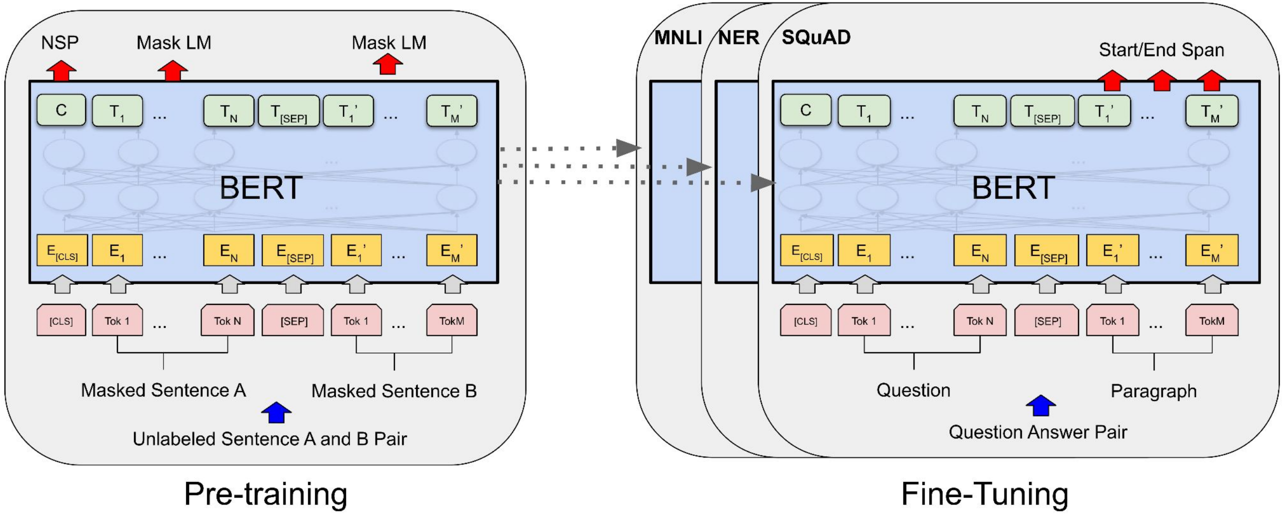
- WordPiece tokenization with a 30,000-token vocabulary
- Maximum sequence length: 512 tokens
- Special tokens: [CLS] at the start, [SEP] between sentences and at the end

Pretraining Corpora:

- BookCorpus (800M words)
- English Wikipedia (2.5B words)

BERT. Use Cases

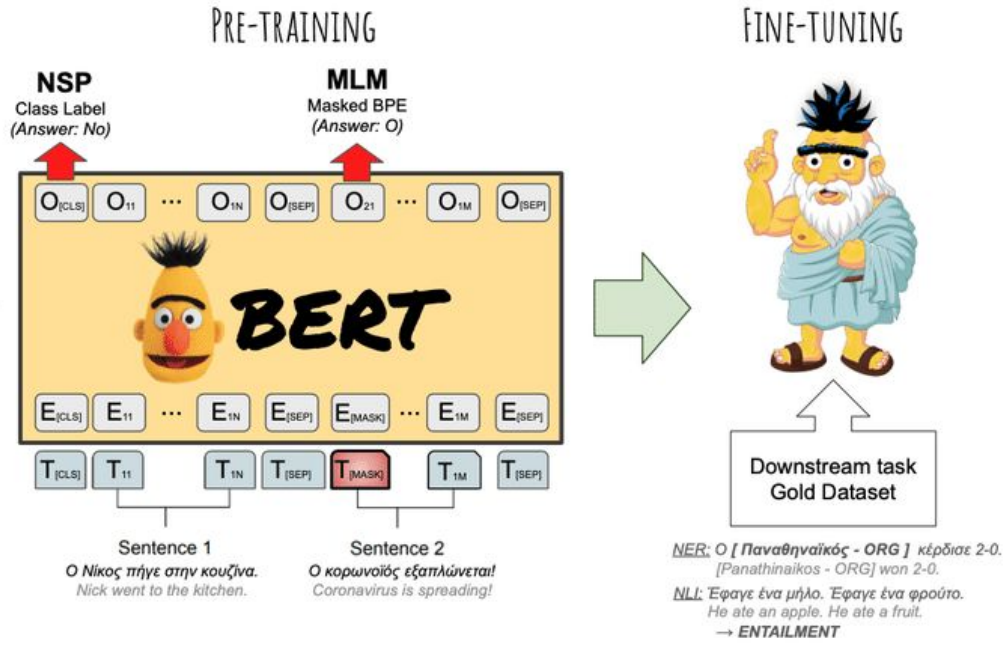
- Context and semantic understanding
- Efficient for text classification and analysis
- Can be fine-tuned for specific tasks (fine-tuning)
- Versions: BERT-base, BERT-large, multilingual BERT



source

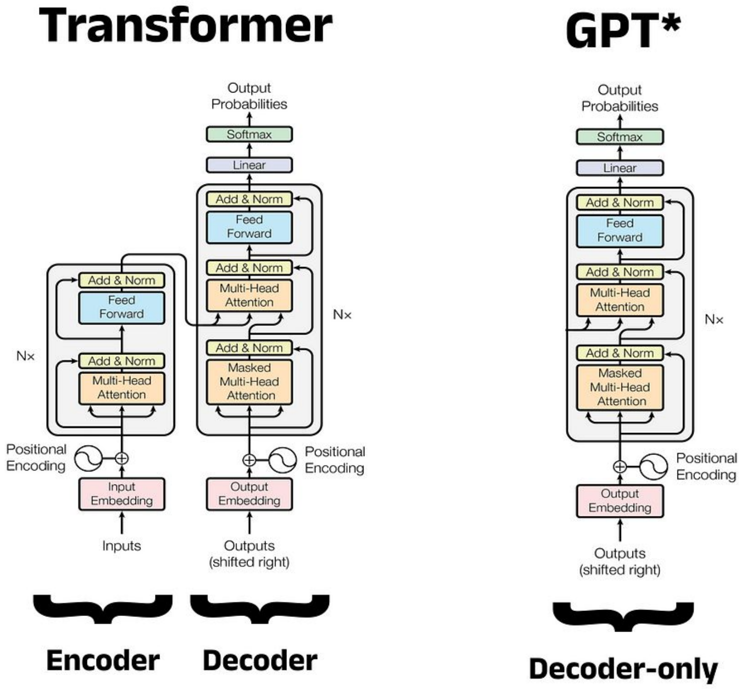
BERT. Usage examples

- Text classification
- Sentiment analysis
- Question answering
- Named entity recognition
- Semantic similarity detection


PRE-TRAINING
FINE-TUNING
NSP
Class Label
(Answer: No)
MLM
Masked BPE
(Answer: O)
$O_{[CLS]}$ O_{11} ... O_{1N} $O_{[SEP]}$ O_{21} ... O_{1M} $O_{[SEP]}$
BERT
$E_{[CLS]}$ E_{11} ... E_{1N} $E_{[SEP]}$ $E_{[MASK]}$ ... E_{1M} $E_{[SEP]}$
$T_{[CLS]}$ T_{11} T_{1N} $T_{[SEP]}$ $T_{[MASK]}$ T_{1M} $T_{[SEP]}$
Sentence 1
Ο Νίκος πήγε στην κουζίνα.
Nick went to the kitchen.
Sentence 2
Ο κορωνοϊός εξαπλώνεται!
Coronavirus is spreading!
Downstream task
Gold Dataset
NER: Ο [Παναθηναϊκός - ORG] κέρδισε 2-0.
[Panathinaikos - ORG] won 2-0.
NLI: Έφαγε ένα μήλο. Έφαγε ένα φρούτο.
He ate an apple. He ate a fruit.
→ ENTAILMENT

GPT

- Generative Pre-trained Transformer
- Based on decoder-only Transformer architecture
- Autoregressive modeling (next-token prediction)
- Trained on massive unlabelled text datasets



GPT-1. Architecture and Training

Architecture:

- Decoder-only transformer
- 12 layers
- 768-dimensional embeddings
- 12 attention heads
- Total parameters: 117M

Training Details:

- Batch size: 64 sequences
- Optimizer: Adam
- Max learning rate: 2.5e-4
- Linear warmup for the first 2000 updates
- Cosine decay down to 0
- Total updates: 100 epochs on 1B tokens (~800k updates)
- Dropout: 0.1 on attention outputs and feed-forward layers
- L2 regularization: 0.01 for non-embedded weights

GPT-1. Corpora and Data.

Input Data:

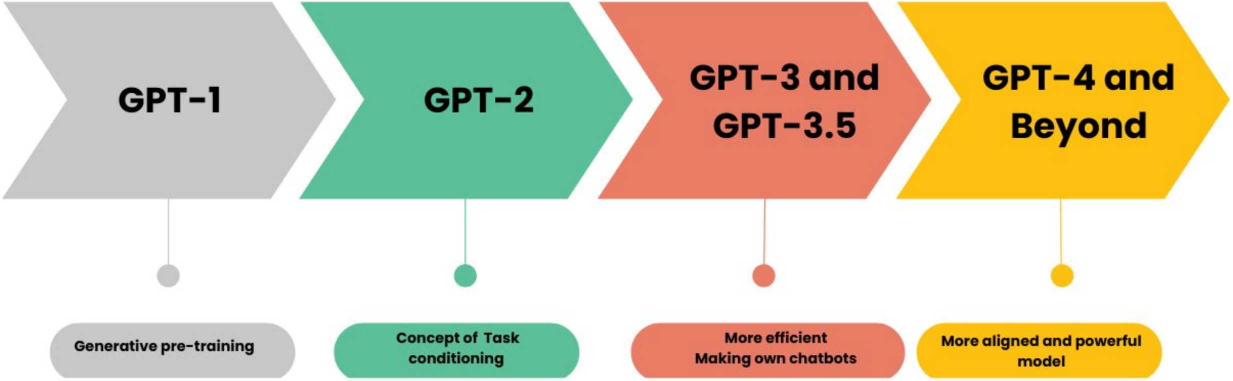
- BytePair Encoding (BPE) with a 40,000-token vocabulary
- Special tokens: [START], [END], [EXTRACT]
- Sequence length: 512 tokens

Pretraining Corpora:

- BookCorpus (over 7,000 unpublished books)

GPT. Model Versions

- **GPT-1 [2018]:** 117M parameters
- **GPT-2 [2019]:** 1.5B parameters, improved text generation quality
- **GPT-3 [2020]:** 175B parameters, few-shot learning
- **GPT-3.5 [2022]:** GPT-3 + Instruct tuning + RLHF
- **GPT-4 [2023]:** Multimodal, improved context understanding

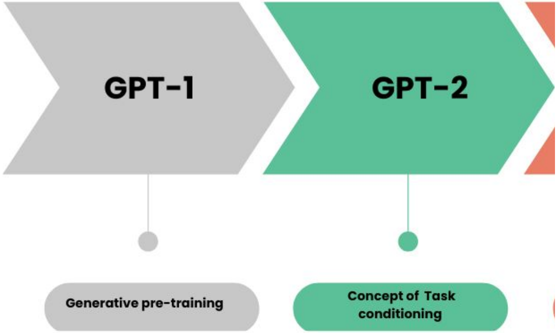

GPT-1
GPT-2
GPT-3 and GPT-3.5
GPT-4 and Beyond
Generative pre-training
Concept of Task conditioning
More efficient Making own chatbots
More aligned and powerful model

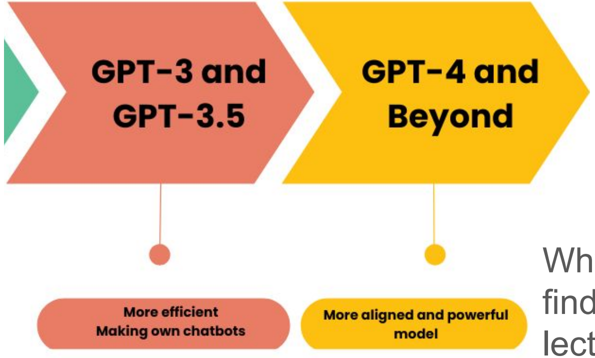
GPT. Usage examples

- Text generation
- Question answering
- Summarization
- Translation
- Code writing
- Content creation (articles, poetry, scripts)

+

- Classification: LLMs can classify texts by simply generating a class label.
- NER: Models can identify named entities by generating them in a specific format.
- Sentiment analysis: Generating a sentiment score for the text.
- Filling in blanks in text.
- Solving mathematical problems.
- Logical reasoning and deduction.

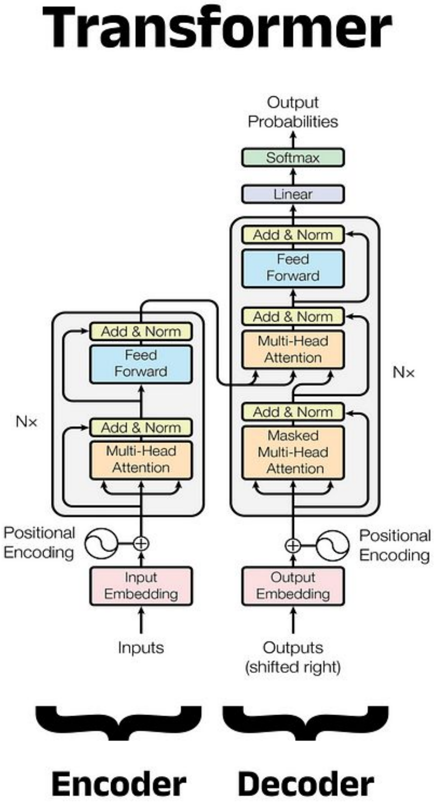



GPT-3 and GPT-3.5
GPT-4 and Beyond
More efficient Making own chatbots
More aligned and powerful model

Why is that? We'll find out in the next lecture.

T5

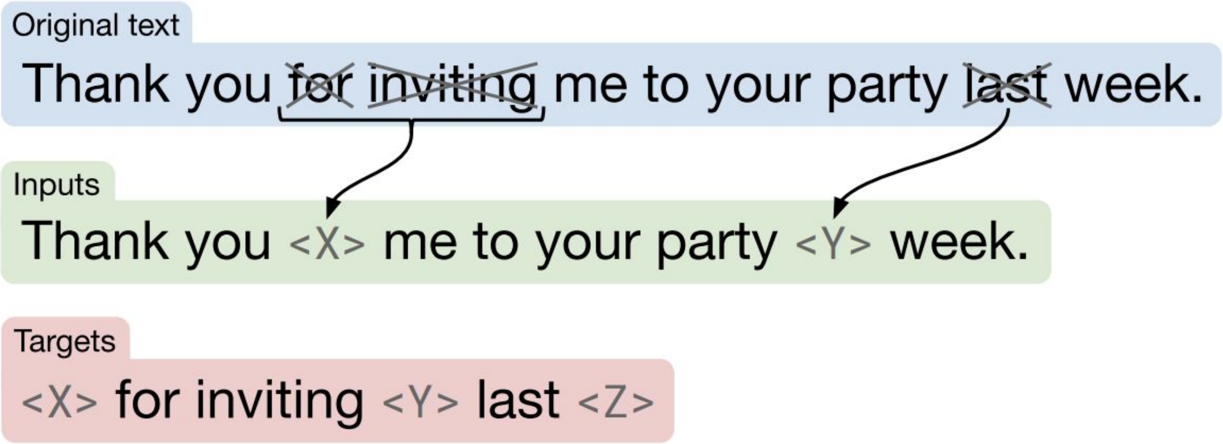
- Text-to-Text Transfer Transformer
- Unified approach: all tasks represented as text-to-text transformation
- Uses encoder-decoder transformer architecture



2019

T5. Pretrain Tasks.

- **Masked Language Modeling** with “span corruption”
- All tasks presented in a text-to-text format



Original text

Thank you ~~for inviting~~ me to your party ~~last~~ week.

Inputs

Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

source

T5. Architecture and Training

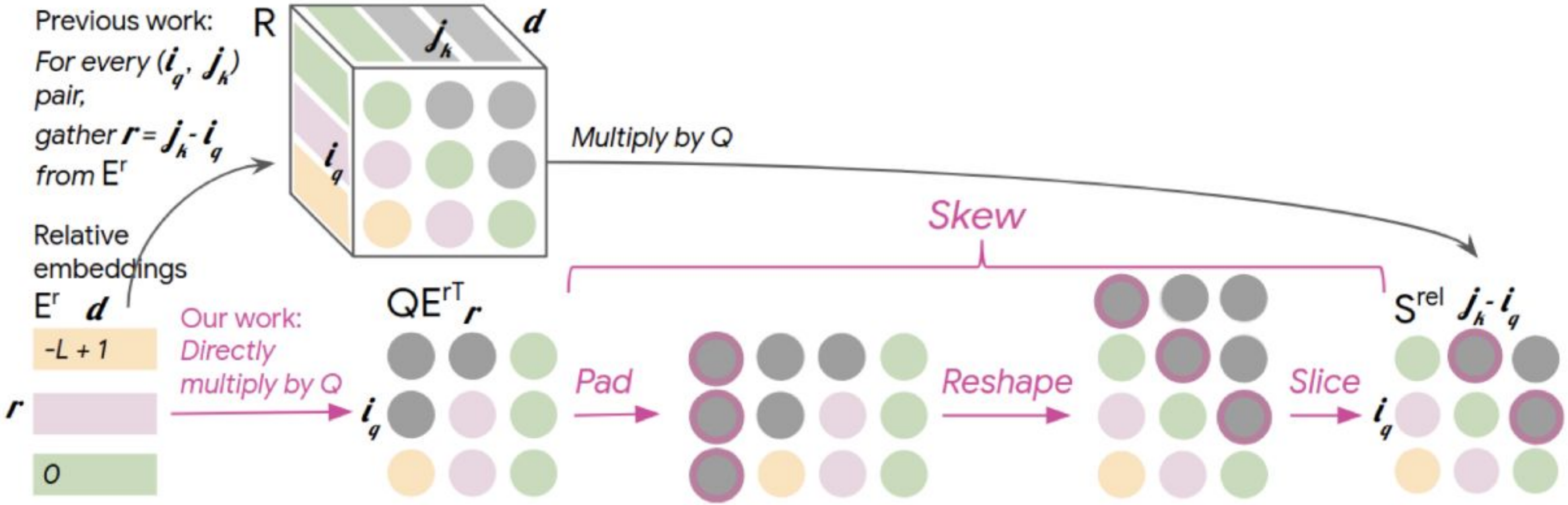
Architecture:

- Encoder-decoder transformer
- Various model sizes (Small, Base, Large, 3B, 11B)
- T5-Base:
  - 12 encoder and 12 decoder layers
  - 768-dimensional embeddings
  - 12 attention heads
- Total parameters: ~220M

Training Details:

- Batch size: 128 sequences
- Optimizer: AdaFactor
- Constant learning rate: 0.01
- Trained on 1 trillion tokens
- Dropout: 0.1
- Using prefixes to denote tasks (for example, 'translate English to German:')
- Using relative positional encoding – a method that employs a power law and sinusoidal functions for effective encoding of relative positions between tokens in a sequence, allowing for better scalability on long texts and improving the model's generalization capability.

T5. Architecture and Training



source

T5. Corpora and Data.

Input Data:

- SentencePiece tokenizer with a 32,000-token vocabulary
- Sequence length: 512 tokens by default, but can be increased
- All tasks are presented as text-to-text transformations
- Use of special tokens to denote the beginning and end of a sequence

Pretraining Corpora:

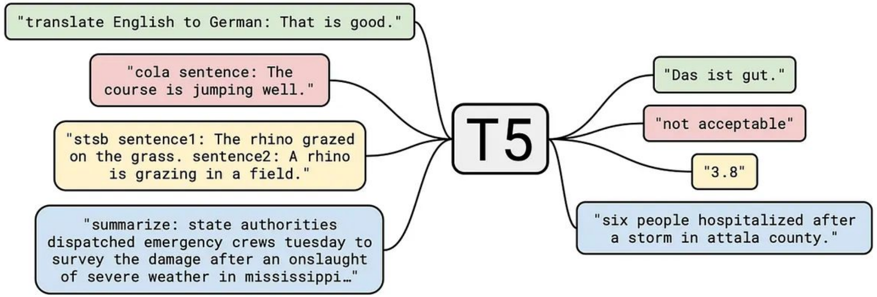
- C4 (Colossal Clean Crawled Corpus)
- Wikipedia
- WebText

T5. "Text-to-Text" principle

- **Input:** text + task prefix
- **Output:** text response
- Unified model for various natural language processing tasks

DEEP (LEARNING) FOCUS

T5: Text-to-Text Transformers (Part One)



"translate English to German: That is good."

"cola sentence: The course is jumping well."

"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

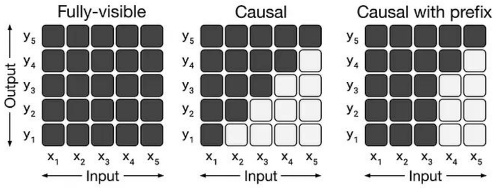
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"

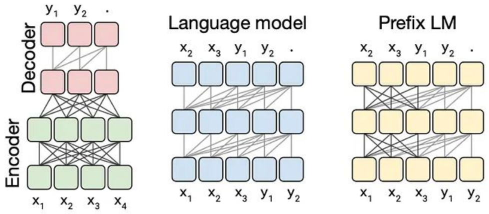
"Das ist gut."

"not acceptable"

"3.8"

"six people hospitalized after a storm in attala county."



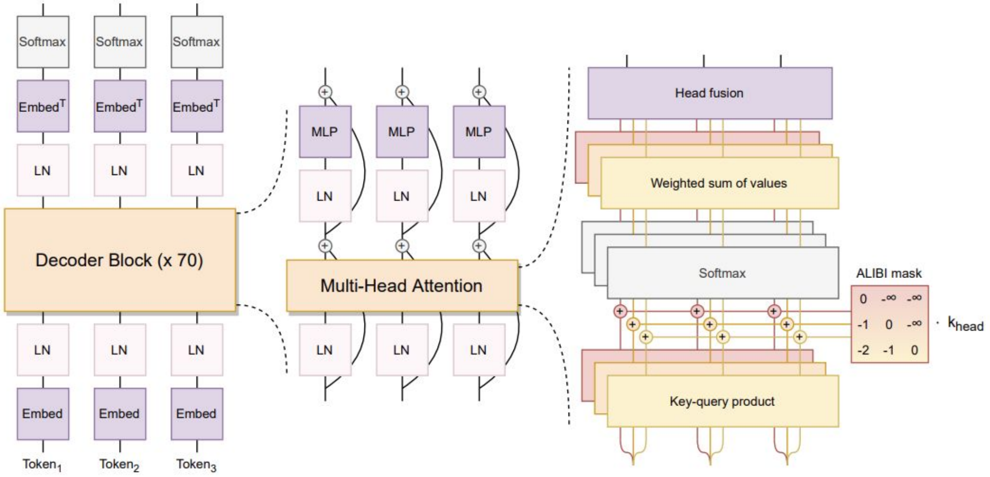


T5. Usage examples

- Machine translation
- Summarization
- Question answering
- Text generation
- Classification
- Paraphrasing

BLOOM

- BigScience Large Open-science Open-access Multilingual Language Model
- Open-access multilingual model with 176B parameters
- GPT-like decoder-only transformer



2022

BLOOM. Technical details of the architecture.

- Support for 46 natural languages and 13 programming languages
- Extended context window
- Fully open-source code and model weights
- Performance optimization for handling a huge number of parameters
- Efficient distributed training

BLOOM. Architecture and Training.

Architecture:

- Decoder-only transformer
- 176B parameters
- 70 transformer layers
- 112 attention heads
- Hidden state dimension: 14,336

Training Details:

- Distributed training on 384 NVIDIA A100 80GB GPUs
- Use of Megatron-DeepSpeed library for optimization
- Use of mixed precision (FP16 and BF16)
- ZeRO (Zero Redundancy Optimizer) technique for memory optimization
- Gradient accumulation to increase effective batch size
- Batch size: 2048 sequences
- Optimizer: AdamW
- Learning rate: initial 6e-5, with cosine decay
- Weight decay: 0.1
- ALiBi (Attention with Linear Biases) embeddings – ALiBi adds a fixed bias to attention scores that linearly decreases with token distance

source

BLOOM. Corpora and Data.

Input Data:

- Uses BPE (Byte-Pair Encoding) tokenizer
- Sequence length: 2048 tokens
- Vocabulary size: 250,680 tokens
- Low-quality content filtering
- Deduplication

Pretraining Corpora:

- ROOTS corpus: 1.6 trillion tokens
- Includes texts in different languages, scientific publications, code
- Trained on 46 natural languages and 13 programming languages
- Supports over 100 languages

PaLM

- Pathways Language Model
- 540 billion parameters
- Based on transformer architecture with optimizations
- Trained on diverse datasets, including web pages, books, GitHub, and social media

2022

PALM. Architecture and Training.

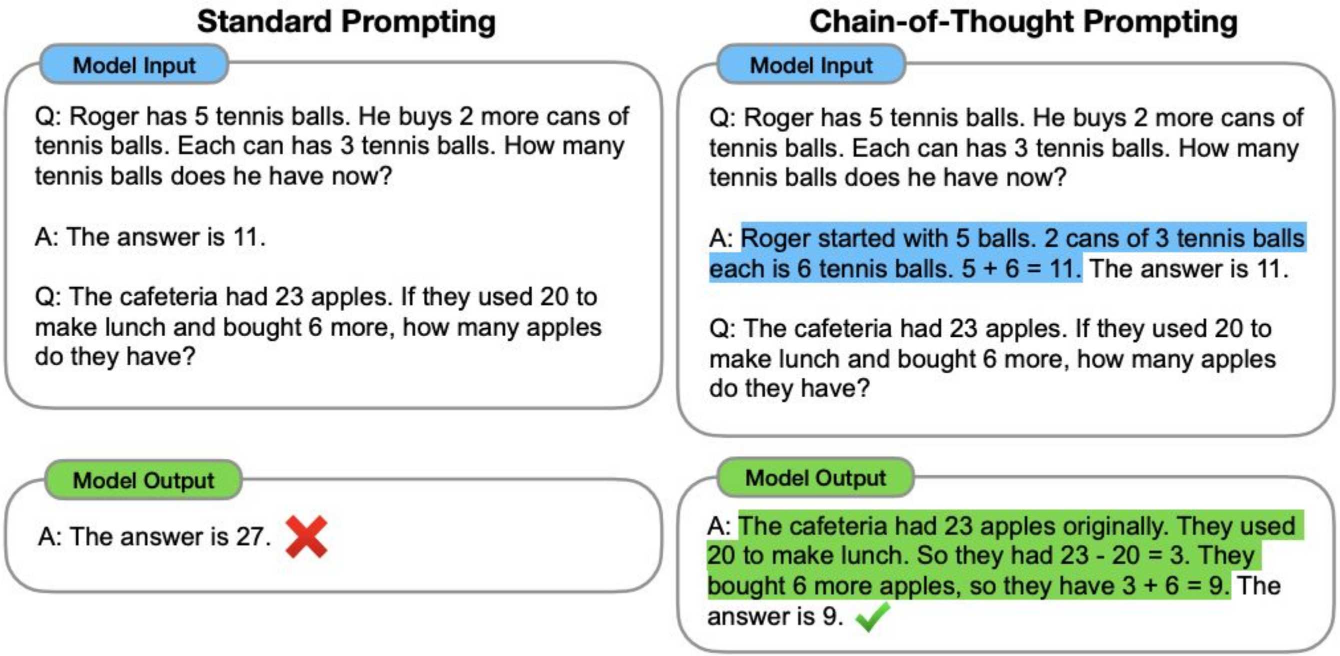
Architecture:

- Decoder-only transformer
- Model sizes: 8B, 62B, 540B parameters

Training Details:

- Uses Pathways technology for distributed training on thousands of TPU chips
- Utilizes SwiGLU instead of ReLU
- Applies RMSNorm instead of LayerNorm
- Capable of handling a wide range of natural language processing (NLP) tasks
- Rotary Position Embedding (RoPE): applies rotations to query and key vectors in the attention mechanism, depending on token position and vector dimension

PALM. Architecture and Training.



Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

source

PALM. Architecture and Training.

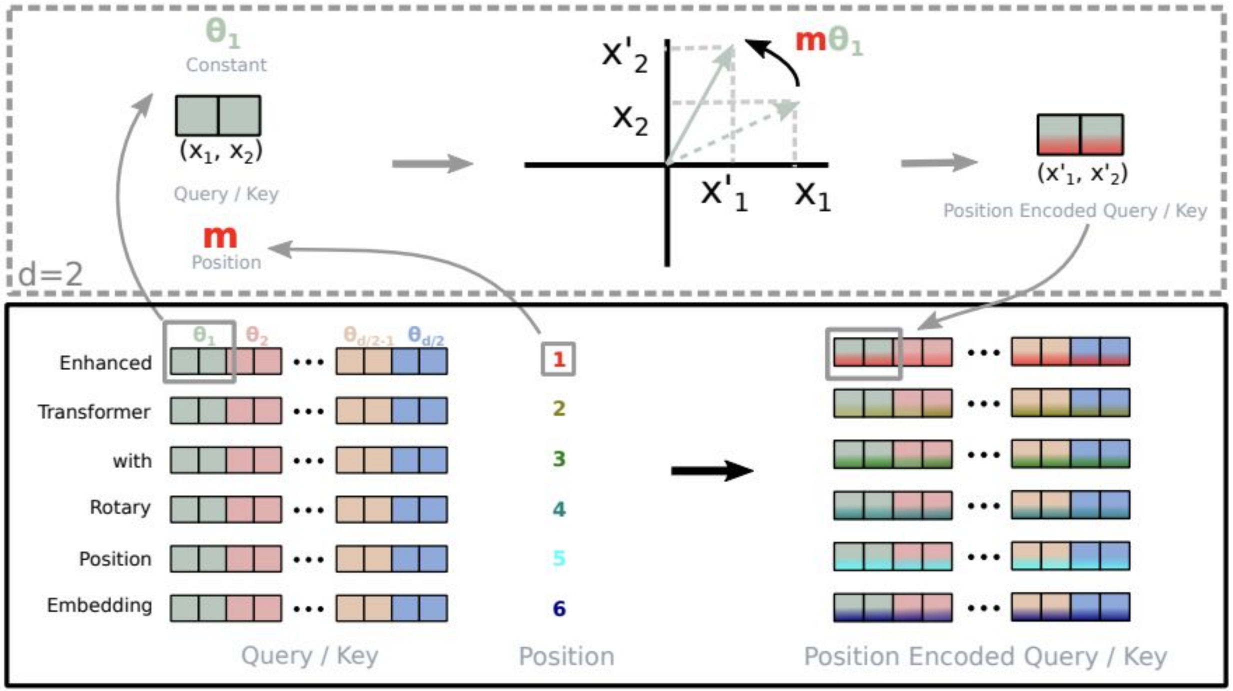


Figure 1: Implementation of Rotary Position Embedding(RoPE).

source

PALM. Corpora and Data.

Input Data:

- SentencePiece tokenizer with a vocabulary of ~256,000 tokens
- Sequence length: 2048 tokens
- Few-shot learning for task adaptation
- Chain-of-Thought Prompting for enhanced reasoning capabilities

Pretraining Corpora:

- Trained on multilingual datasets, including web pages, books, articles, social media, and code
- Data filtering to improve the quality of the training set

PALM. Applications and Features

- **Multitasking and Few-shot Learning:** Capable of adapting to new tasks with minimal examples
- **Improved Reasoning:** Enhanced ability to solve complex tasks and generate code
- **Broad Use Cases:** Applicable across various NLP fields

FLAN-T5

- FLAN-T5 (Fine-tuned Language Net based on T5)
- Based on the T5 architecture with additional instruction-based training
- Improved performance on tasks requiring instructions

FLAN-T5. Architecture and Training.

Архитектура:

- Encoder-decoder трансформер
- Существует несколько версий: Small (60M параметров), Base (250M), Large (780M), XL (3B), XXL (11B)
- T5-Large имеет 24 слоя (12 в энкодере и 12 в декодере)
- T5-Large имеет 16 голов внимания в каждом слое
- Размерность эмбеддинга: 1024
- Размер скрытого состояния: 1024

Детали обучения:

- Обучена на множестве задач с использованием инструкций (prompt-based learning)
- Использует T5's "span corruption" предобучение
- Применяет технику "instruction tuning"
- Использует GeLU
- Применяет LayerNorm
- Использование относительного позиционного кодирования
- Обучена на более широком наборе задач, чем оригинальная T5
- Batch: 32/64

source

FLAN-T5. Corpora and Data.

Input Data:

- SentencePiece tokenizer with a vocabulary of ~32,000 tokens
- Sequence length: 512 tokens (can be increased)

Pretraining Corpora:

- T5 datasets +
- Natural Language Understanding (NLU) tasks: SuperGLUE, SQuAD, etc.
- Text generation tasks: CNN/Daily Mail for summarization, WMT for machine translation
- Multilingual datasets: XNLI, PAWS-X
- Reasoning tasks: ANLI, LogiQA
- Mathematical tasks
- Coding tasks: CodeXGLUE, HumanEval

FLAN-T5. Usage examples

- Contextual question answering
- Text generation from instructions
- Solving reasoning tasks
- Multilingual NLP tasks