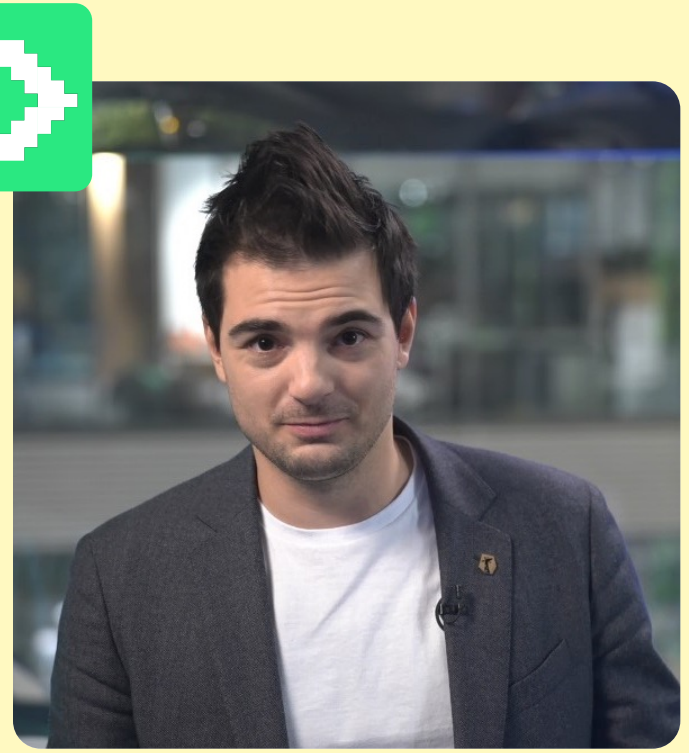
Развитие языковых моделей: BERT, GPT, T5

YOUNG &78 YANDEX

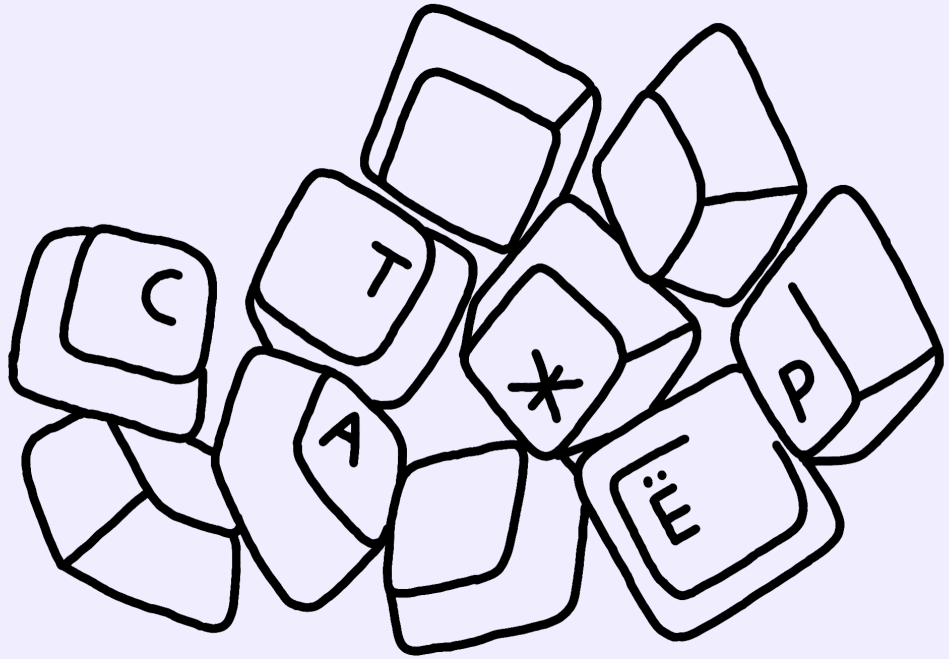


Радослав Нейчев
Выпускник и преподаватель ШАД и МФТИ,
руководитель группы ML-разработки в Яндексе,
основатель girafe-ai

Содержание

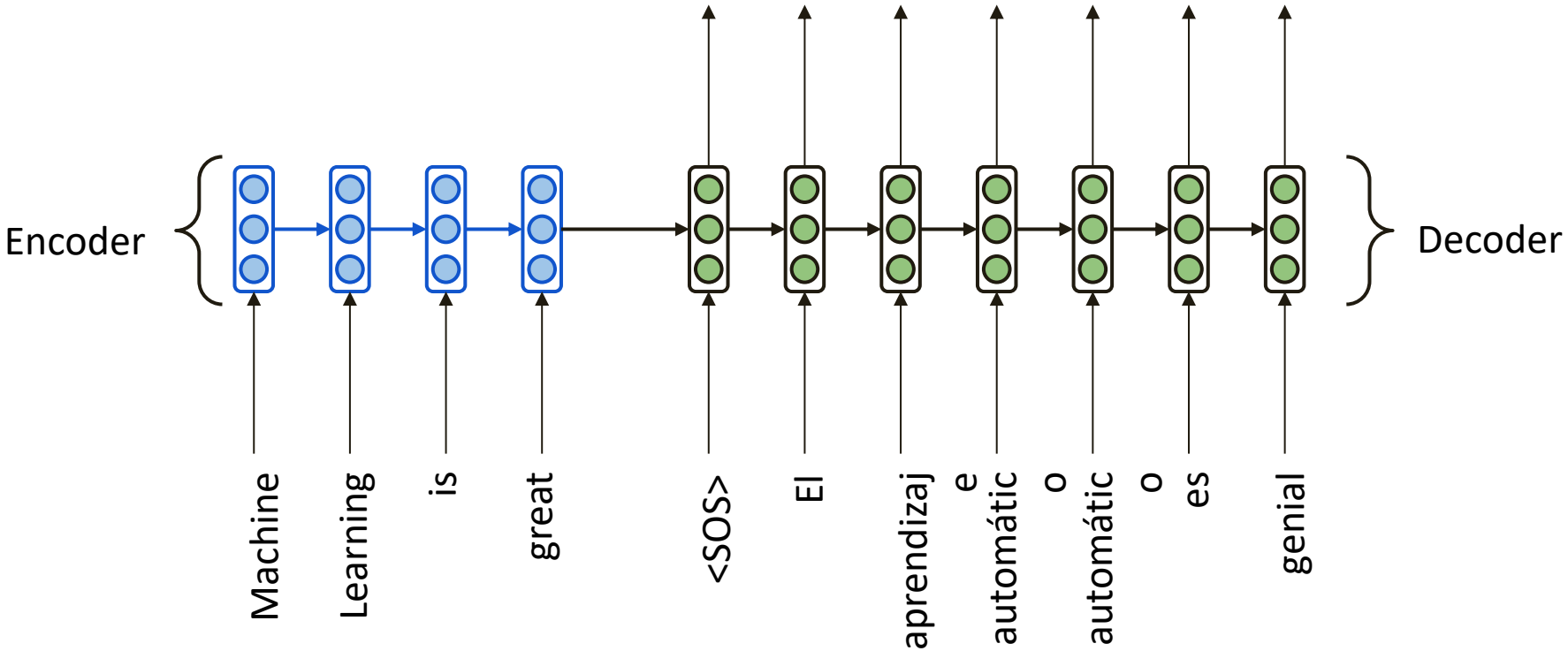
Credits: Слайды составлены в соавторстве с Кариной Романовой

История развития языковых моделей



01

Encoder-decoder архитектура

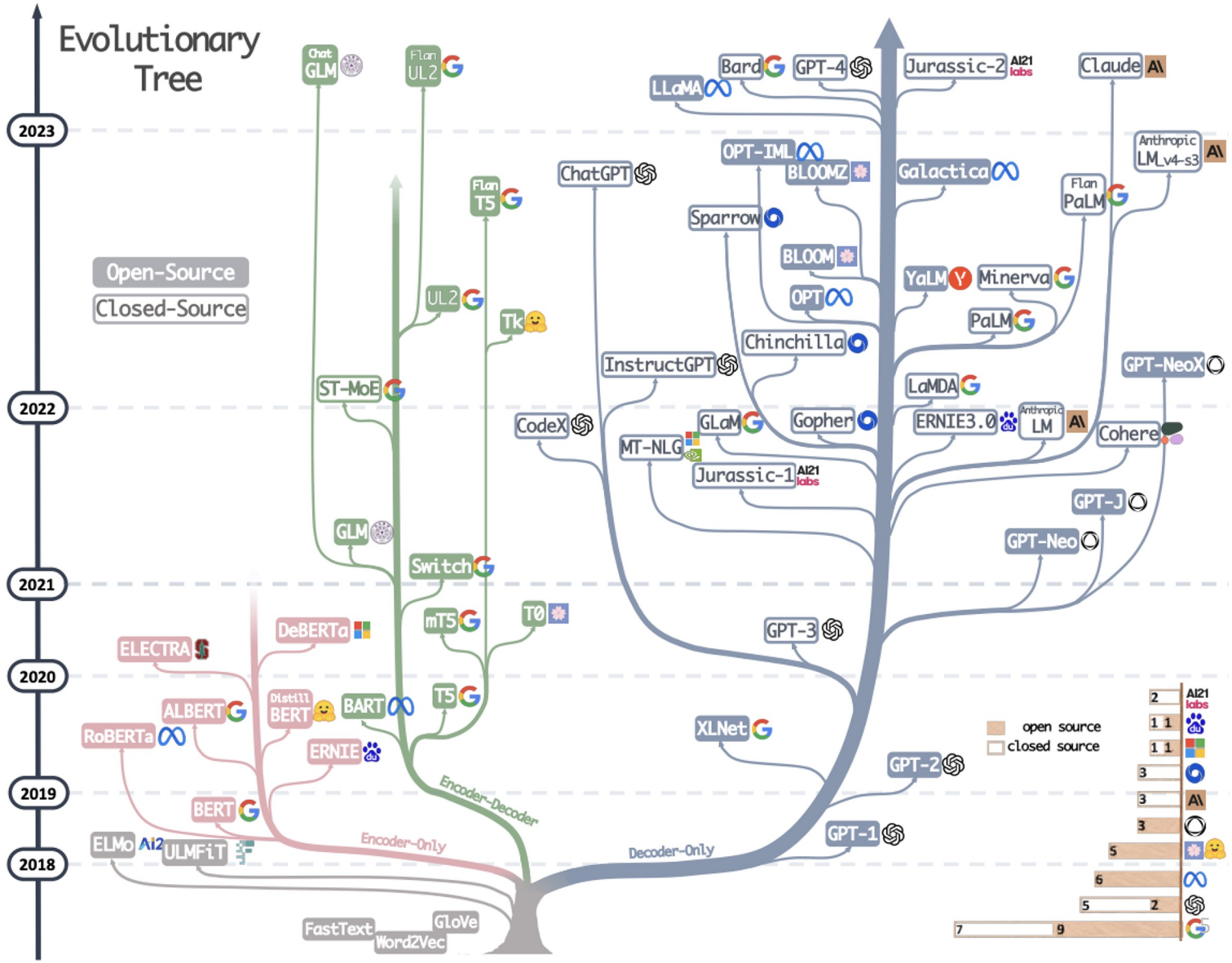


History of Language Model Development

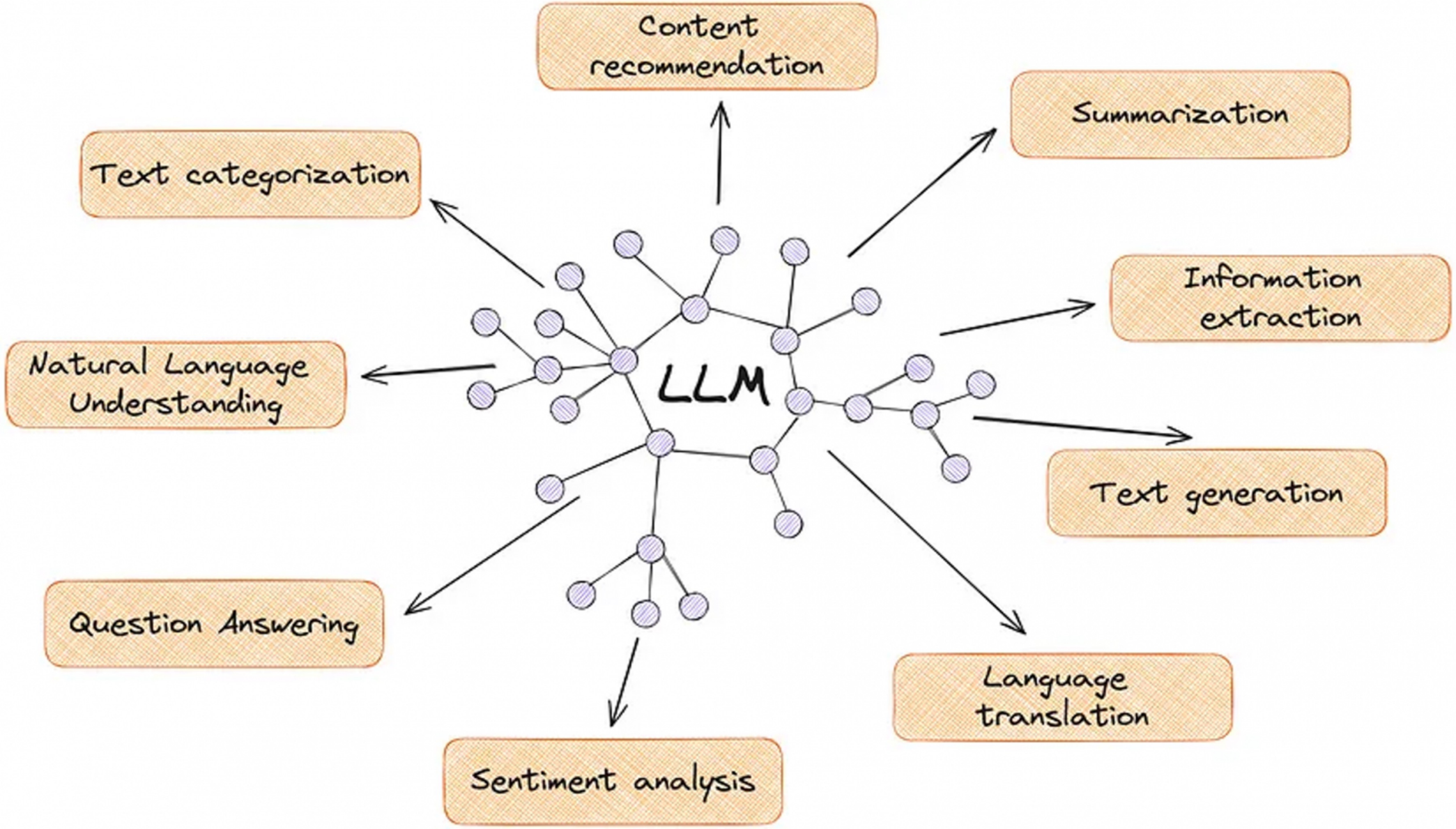
- Decoder-only models (blue branch)
- Encoder-only models (pink branch)
- Encoder-decoder models (green branch)

- Vertical axis shows release dates

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



source: https://arxiv.org/pdf/2304.13712



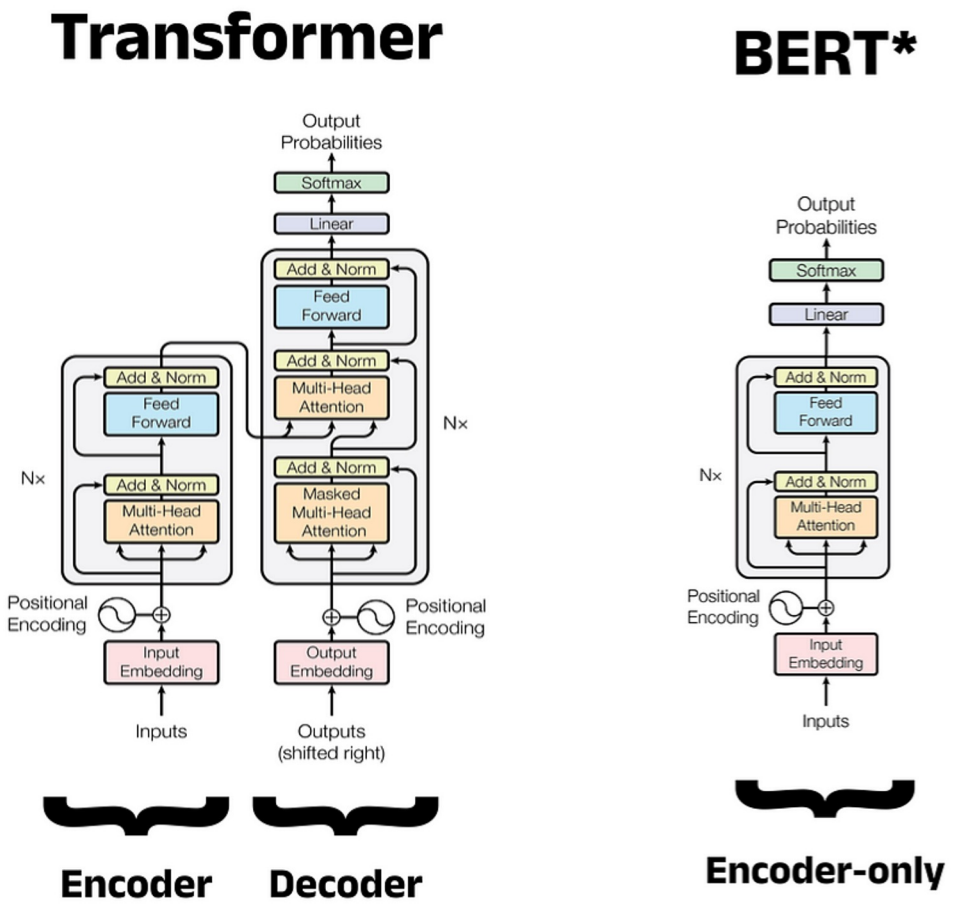
source: https://towardsdatascience.com/all-you-need-to-know-to-develop-using-large-language-models-5c45708156bc

BERT – Bidirectional Encoder Representations from Transformers

02

BERT

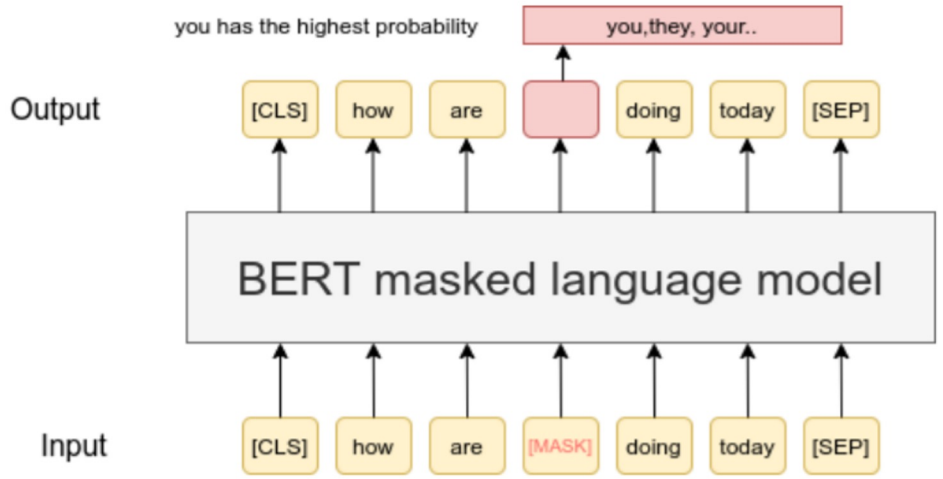
- Encoder-only
- Bidirectional Encoder Representations from Transformers (MLM task)
- Based on Transformer architecture
- Bidirectional context encoding
- Pretrained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks

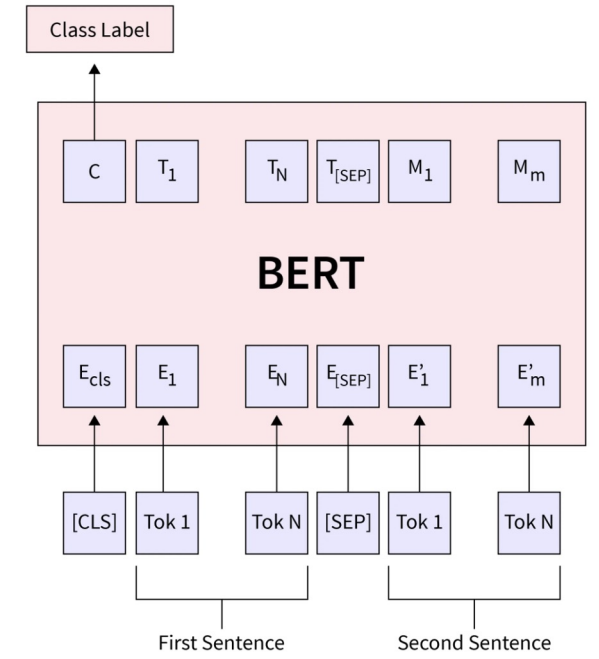


2018

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: https://www.sbert.net/examples/unsupervised_learning/MLM/README.html
source2: https://www.scaler.com/topics/nlp/bert-next-sentence-prediction/

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
  - $\beta 1 = 0.9$
  - $\beta 2 = 0.999$
- Learning rate: 1e-4
- Dropout: 0.1 across all layers
- Activation: GELU

BERT: Corpora and Data

Input Data:

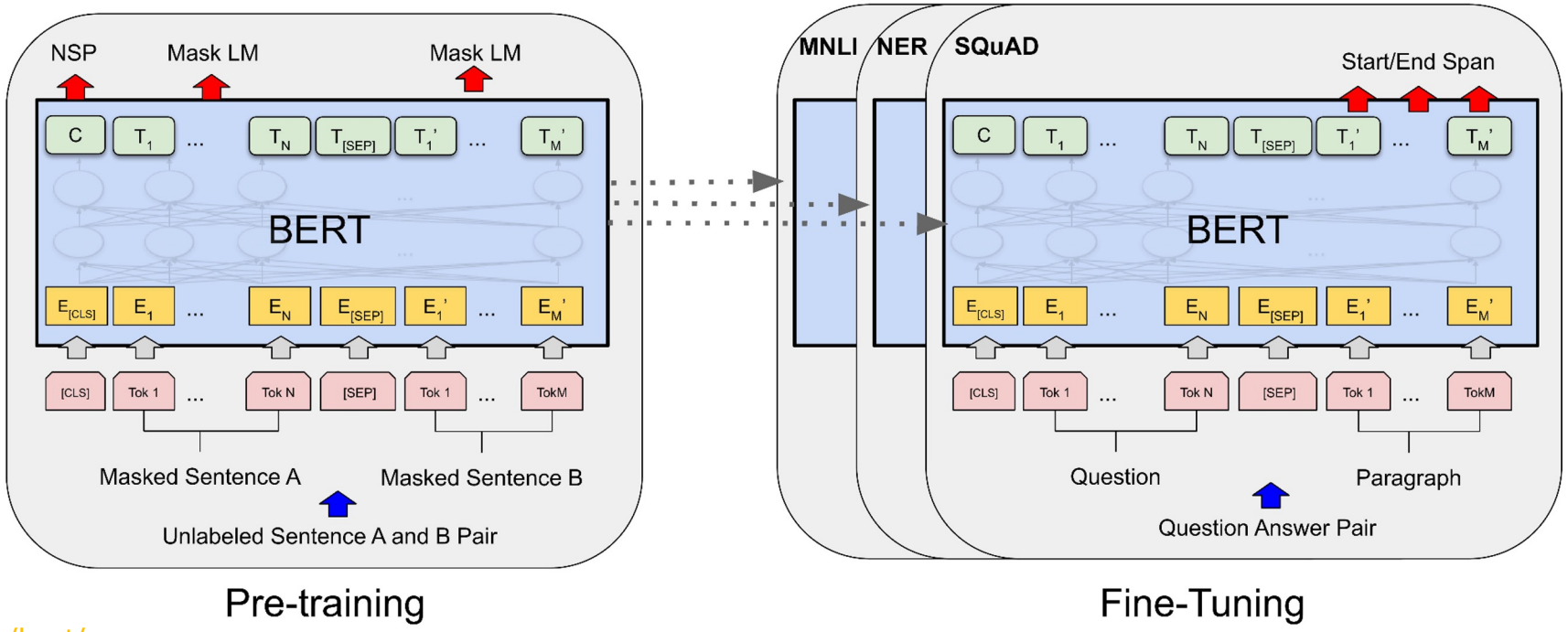
- WordPiece tokenization with a 30,000-token vocabulary
- Maximum sequence length: 512 tokens
- Special tokens: [CLS] at the start, [SEP] between sentences and at the end

Pretraining Corpora:

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

BERT: Use Cases

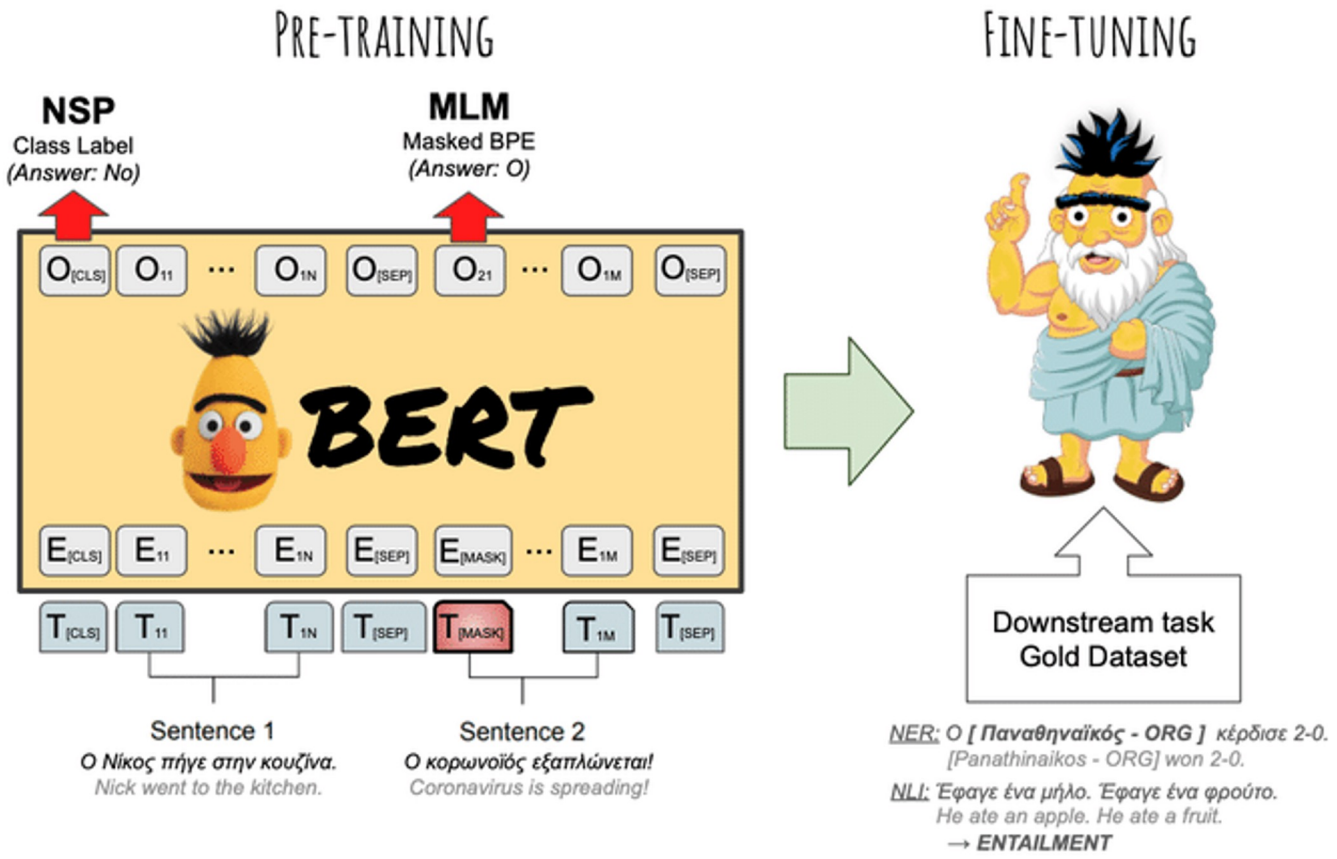
- Context and semantic understanding
- Efficient for text classification and analysis
- Can be fine-tuned for specific tasks (fine-tuning)
- Versions: BERT-base, BERT-large, multilingual BERT



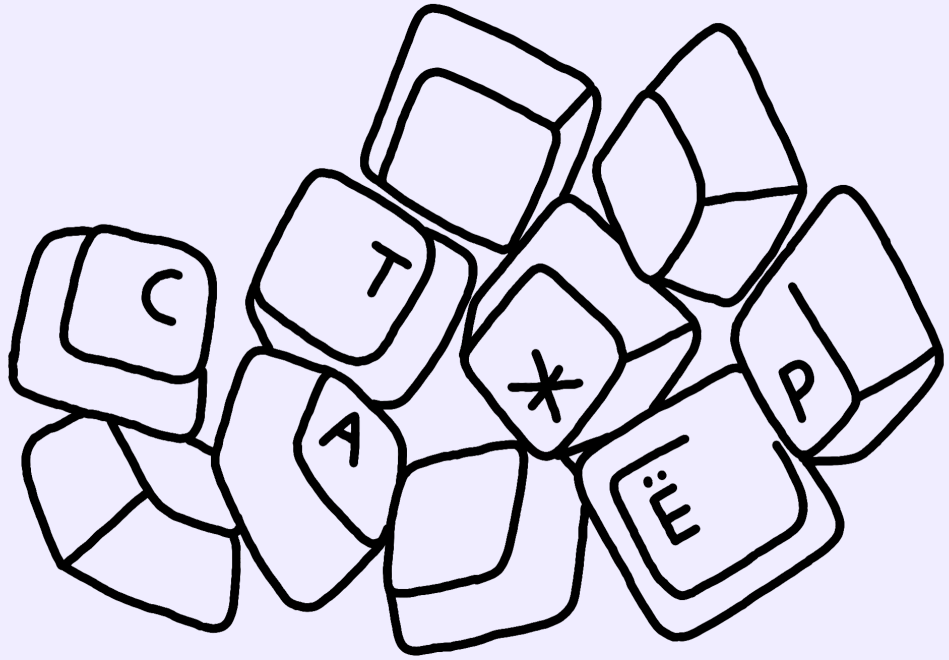
source: https://storrs.io/bert/

BERT: Usage examples

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



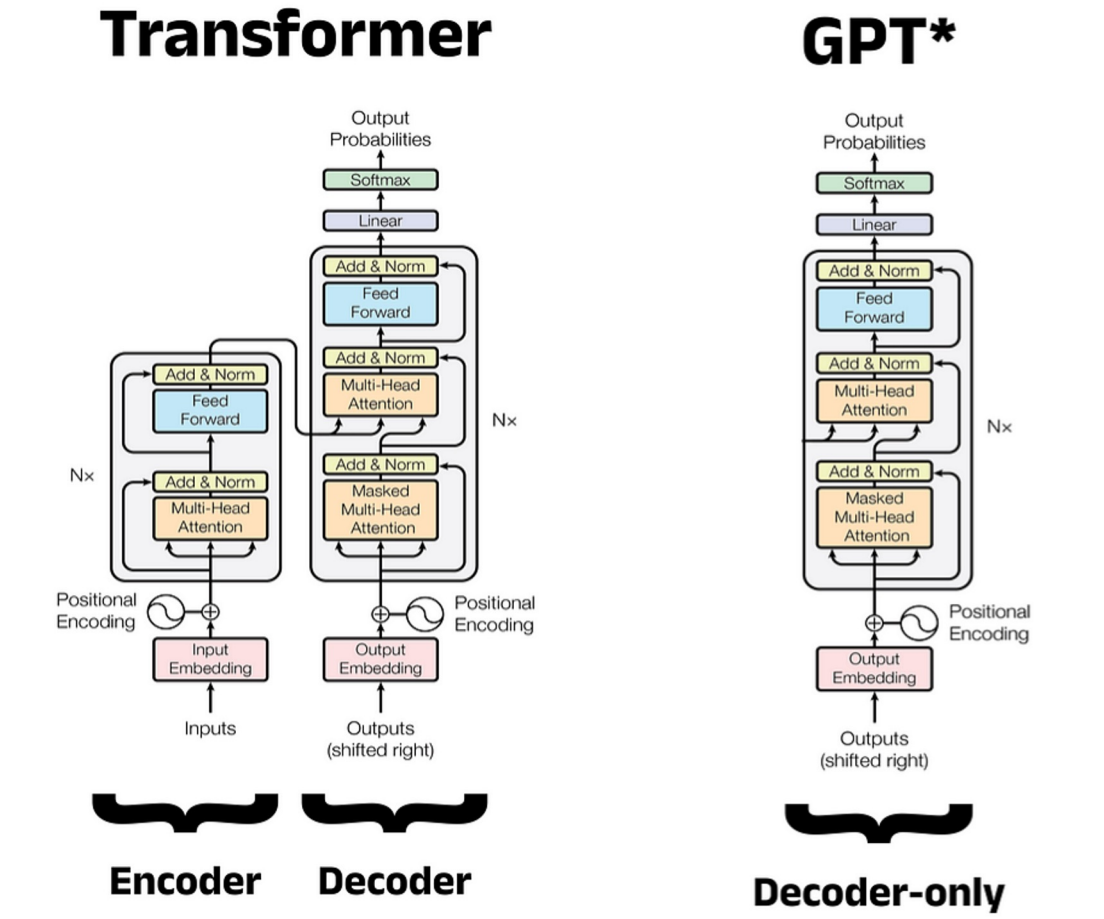
GPT – Generative Pre-trained Transformer



03

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:

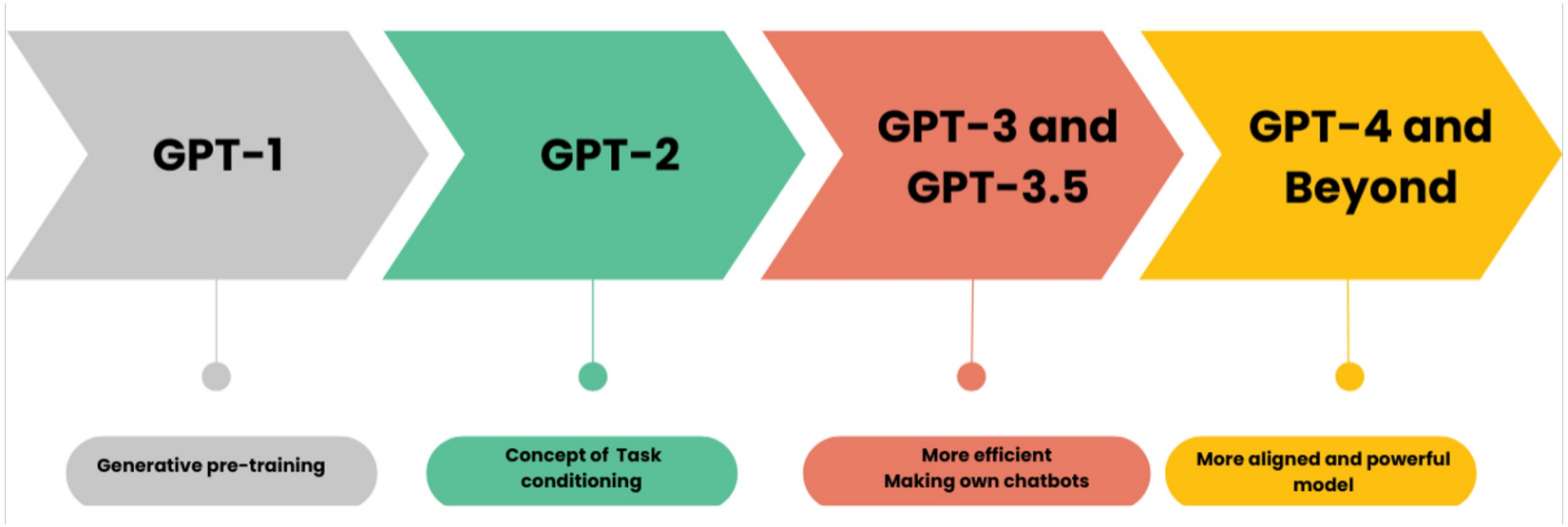
- BytePair Encoding (BPE) with a 40,000-token vocabulary
- Special tokens: [START], [END], [EXTRACT]
- Sequence length: 512 tokens

Pretraining Corpora:

- BookCorpus (over 7,000 unpublished books)

GPT: Model Versions

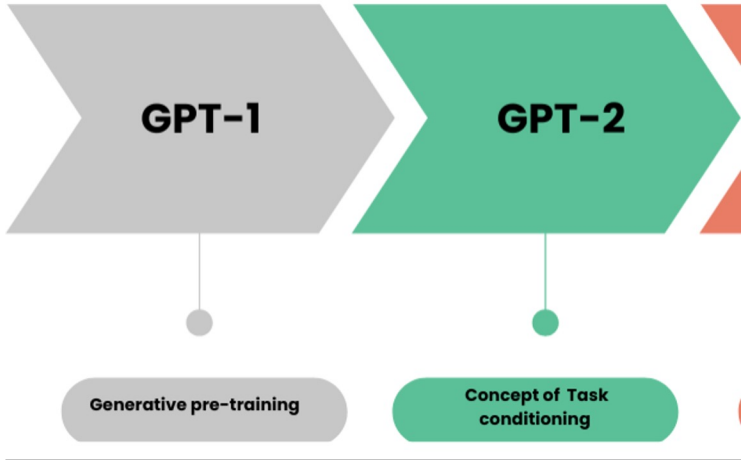
- **GPT-1 [2018]:** 117M parameters
- **GPT-2 [2019]:** 1.5B parameters, improved text generation quality
- **GPT-3 [2020]:** 175B parameters, few-shot learning
- **GPT-3.5 [2022]:** GPT-3 + Instruct tuning + RLHF
- **GPT-4 [2023]:** Multimodal, improved context understanding

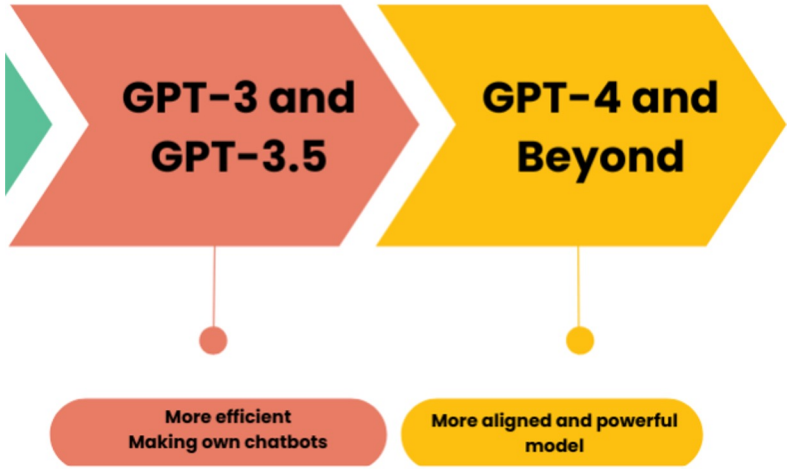


GPT: Usage examples

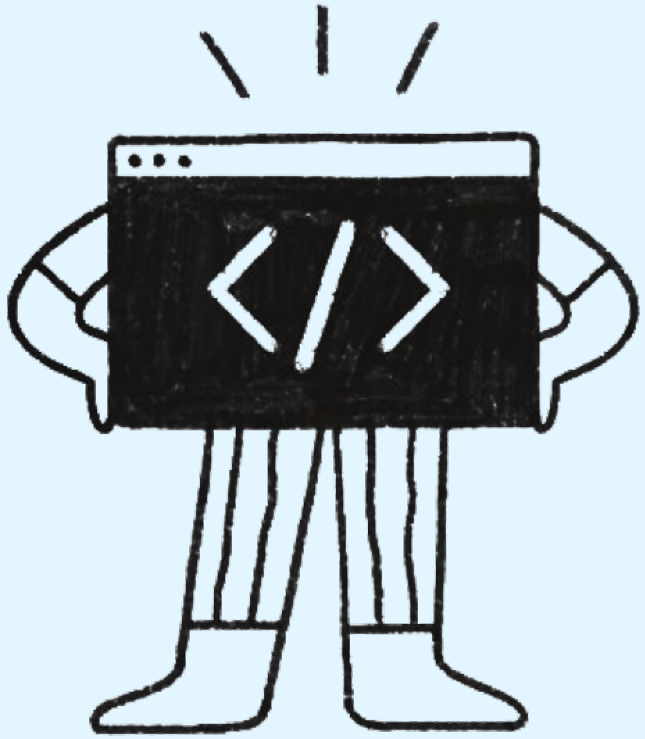
- Text generation
- Question answering
- Summarization
- Translation
- Code writing
- Content creation (articles, poetry, scripts)

- 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





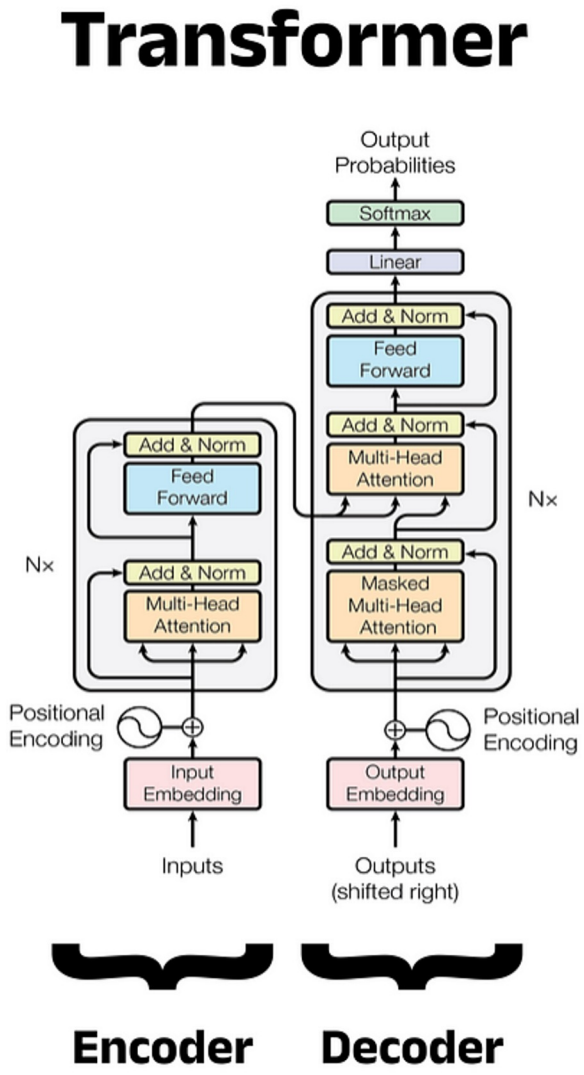
T5 – Text-to-Text Transfer Transformer



04

T5

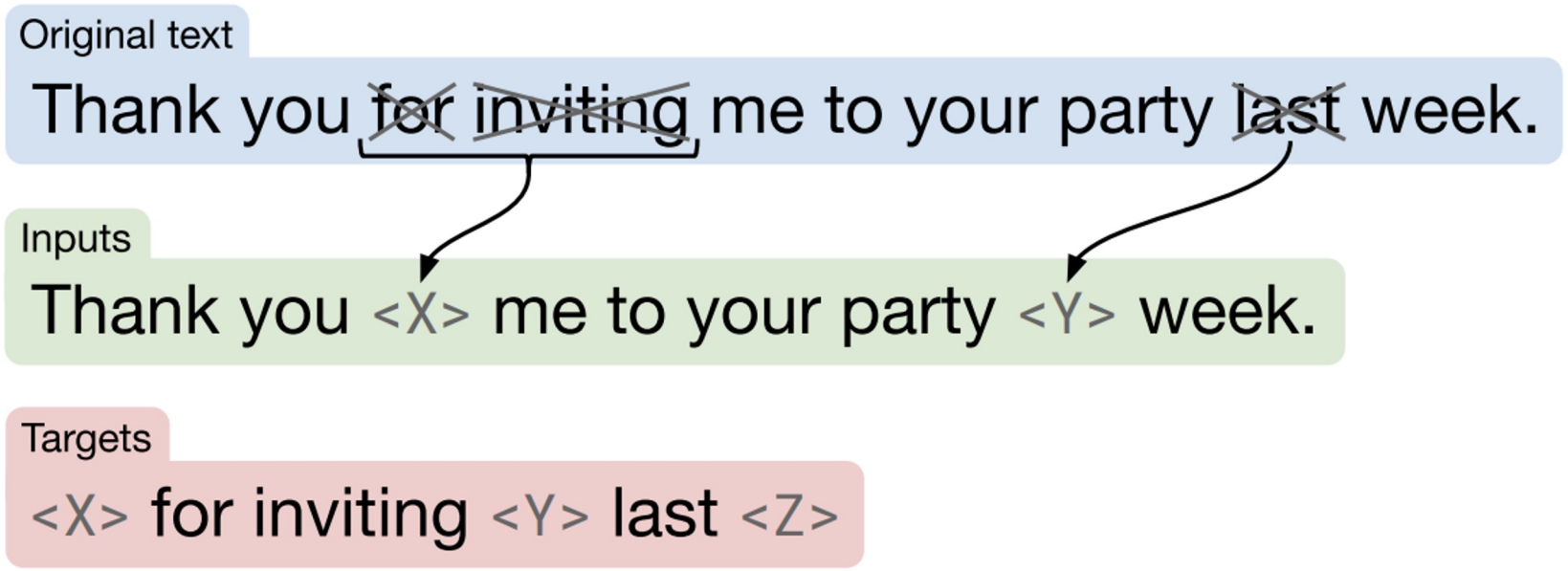
- Text-to-Text Transfer Transformer
- Unified approach: all tasks represented as text-to-text transformation
- Uses encoder-decoder transformer architecture



2019

T5: Pretrain Tasks

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



source: https://arxiv.org/pdf/1910.10683

T5: Architecture and Training

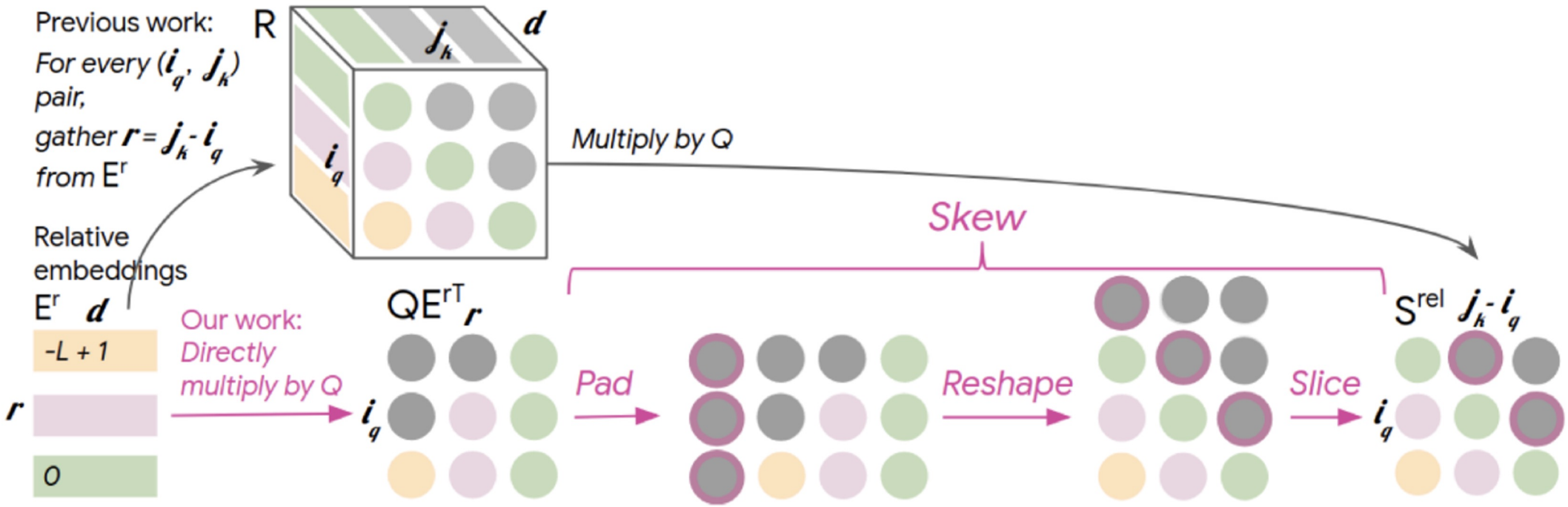
Architecture:

- 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: https://jaketae.github.io/study/relative-positional-encoding/

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

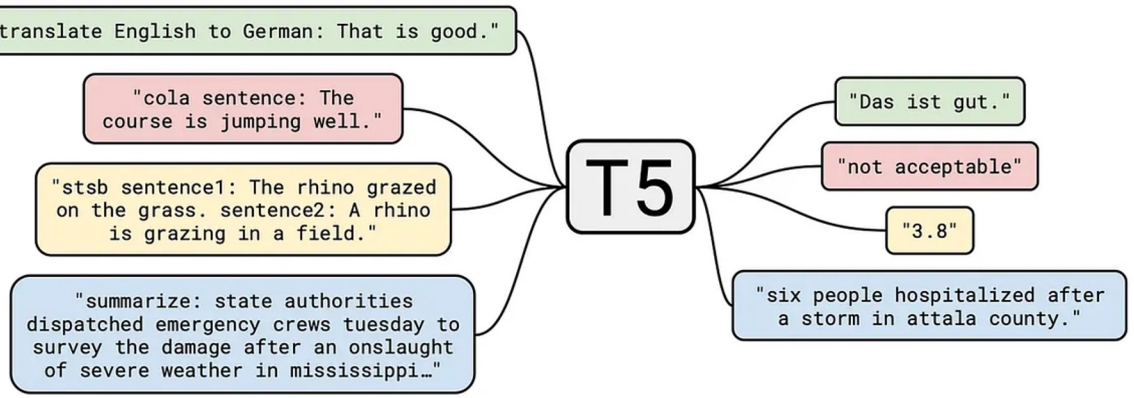
Pretraining Corpora:

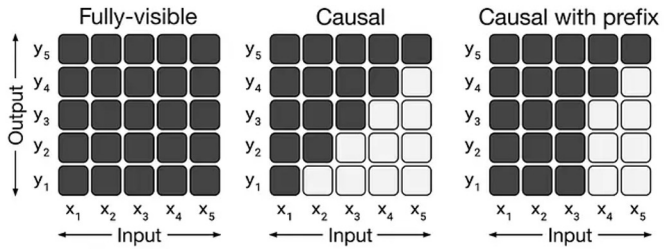
- C4 (Colossal Clean Crawled Corpus)
- Wikipedia
- WebText

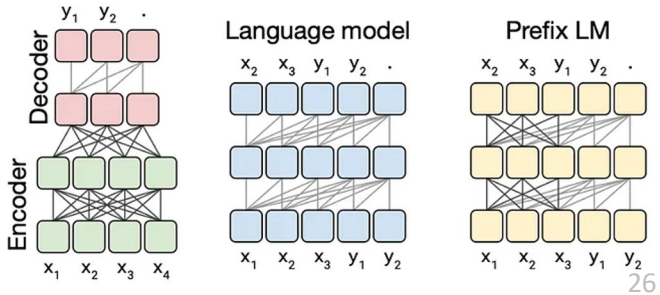
T5: "Text-to-Text" principle

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

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





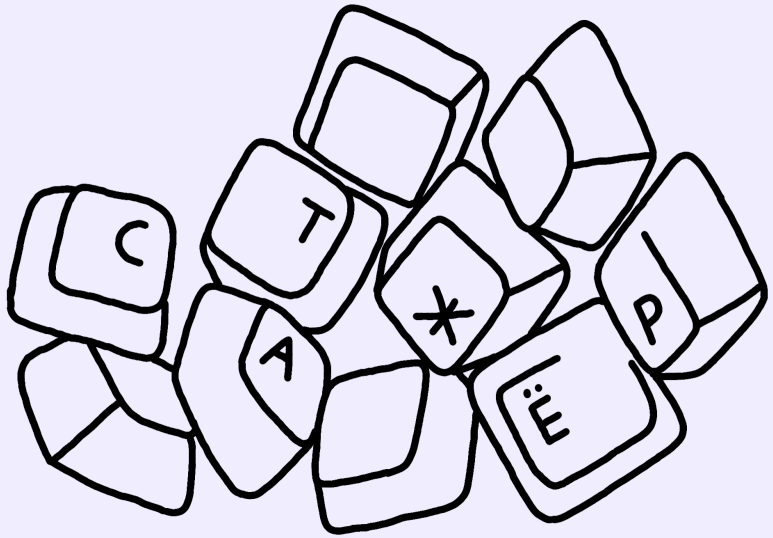


T5: Usage examples

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

Outro про LLM

05



Спасибо за внимание

Y&Y