



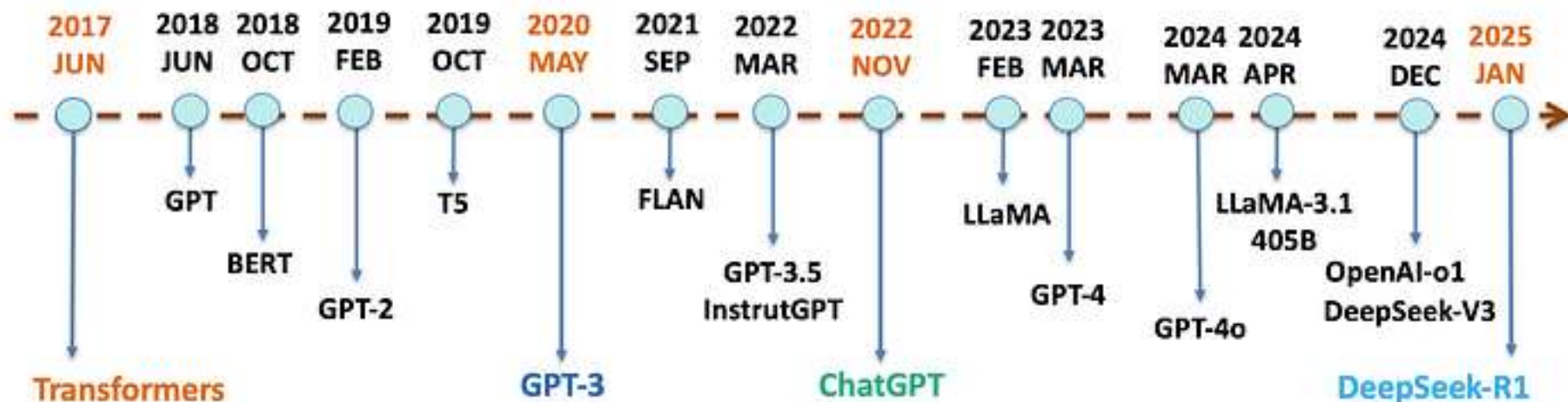
Тверской
государственный
технический
университет

Интеллектуальные информационные системы

Языковые модели

2025 г.

A Brief History of LLMs

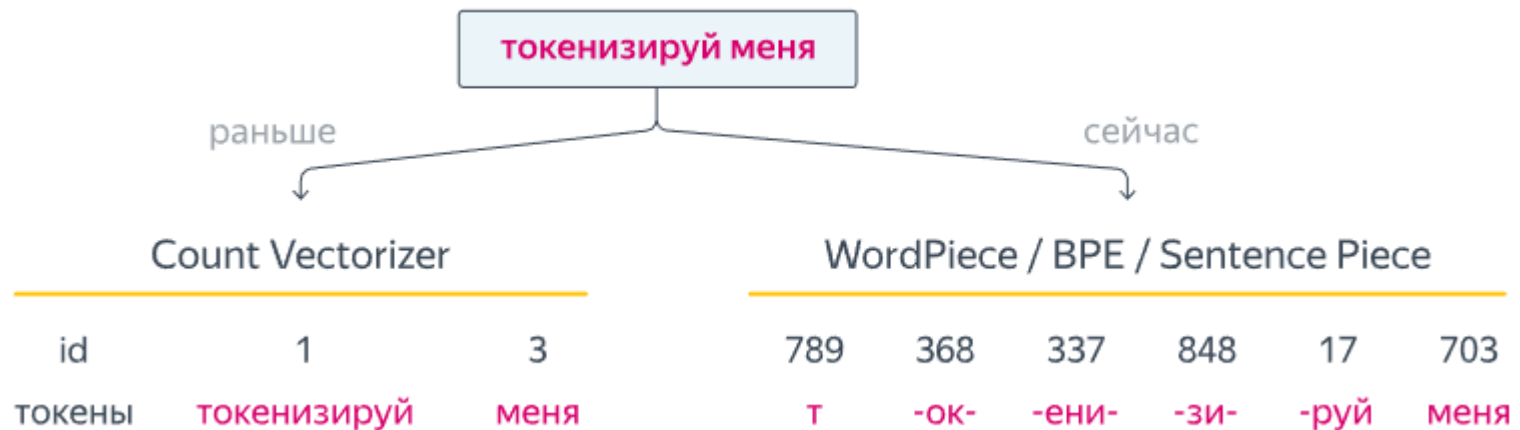


Языковые модели

- Языковые модели (ЯМ) — это системы искусственного интеллекта, предназначенные для обработки, понимания и генерации текста, максимально приближенного к человеческому. Они обучаются на больших объёмах текстовых данных, выявляя закономерности, грамматические структуры и семантические связи. Благодаря этому такие модели могут создавать связный, логичный и контекстуально уместный текст.
- Большая языковая модель (БЯМ, LLM) — языковая модель, основанная на нейронной сети с множеством параметров (миллиарды весовых коэффициентов и более), которая проходит предварительное обучение на обширных массивах неразмеченного текста методами самообучения (обучения с псевдометками, созданными самой моделью, а не внешним учителем), а затем подвергается тонкой настройке (fine-tuning) с применением обучения с подкреплением на основе отзывов людей (RLHF – reinforcement learning from human feedback) для согласования результатов генерации с человеческими предпочтениями (alignment problem) и инструкциями. Термин «большой» в названии больших языковых моделей характеризует два ключевых аспекта: количество параметров и объём обучающих данных. Параметры представляют собой переменные и веса, используемые моделью для формирования прогнозов при обработке языка.

Токенизация

- Текст (документ) – последовательность токенов
- Токенизация – представление текста в виде последовательности
- Токен – атомарный элемент последовательности (слово, символ, морфема (значимая часть слова) и т.д.)



Токенизация

Современные токенизаторы построены на основе алгоритма Byte Pair Encoding (BPE) – итеративного метода subword токенизации.

Алгоритм строит таблицу слияний (merge table) и словарь токенов (vocabulary). Изначально словарь содержит все отдельные символы (буквы, цифры, знаки препинания и т.д.), а таблица слияний пустая. Каждое слово из обучающего корпуса разбивается на последовательность отдельных символов.

Фиксируется размер словаря и выполняются итеративные шаги:

- Подсчет пар: анализируется вся обучающая выборка, считается сколько раз соседняя пара символов встречается в данных.

HELLO -> ['h', 'e', 'l', 'l', 'o'] -> пары ('h','e'), ('e','l'), ('l','l'), ('l','o')

- Выбор самой частой пары: среди всех пар выбирается та, которая чаще всего встречается в обучающей выборке
- Слияние: эта пара объединяется в новый токен, который добавляется в словарь. В таблицу слияний записывается правило вида «всегда объединяй эти два символа в один»

Шаги итеративно повторяются до достижения желаемого размера словаря, в результате модель получает гибкий словарь, содержащий как частые целые слова, так и элементы слов (subwords), что позволяет эффективно обрабатывать как известные, так и редкие и даже никогда не встречавшиеся слова.

AABABVCABBAABAC

ADDCOBADAC

EDCOBEAC

AA - 2

AB - 4 AB = D

BA - 3

BC - 1

CA - 1

BB - 1

AC - 1

AD - 2 AD = E

DD - 1

DC - 1

CD - 1

DB - 1

DA - 1

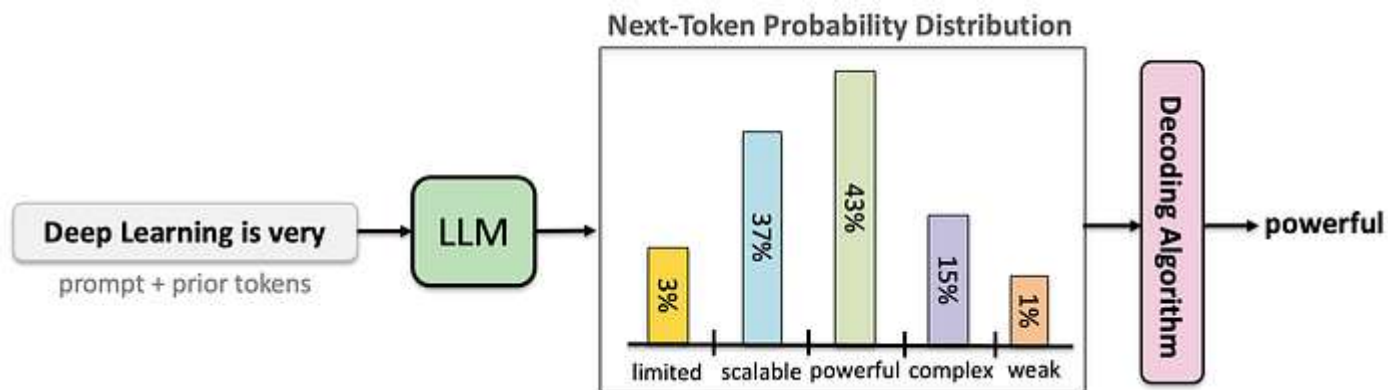
AC - 1

Статистические модели

Статистическая языковая модель — оценивает вероятность последовательности слов в языке на основе статистических закономерностей, выявленных в обучающем корпусе текстов. Такая модель аппроксимирует совместное распределение слов в предложении, используя условные вероятности, часто с ограничением длины контекста. Основная цель модели — предсказать следующее слово в последовательности или оценить правдоподобие заданной последовательности слов, опираясь исключительно на частоты и совместные встречаемости слов в данных.

В выражении $P(w_{t+1}|w_1, w_2, \dots, w_t)$ длина контекста равна t — редко используется длина контекста более 3, потому что:

- Вычислительная сложность. Для каждого возможного уникального контекста длины K при корпусе текста из N различных слов стоимость хранения счетчиков встречаемости для выбранной длины контекста равна N^K
- Достаточность статистик падает с ростом длины контекста, потому что длинный контекст реже встречается в текстах.



Статистические модели

Достоинства статистических моделей:

- Простота реализации
- Высокая скорость работы алгоритма
- Низкая вычислительная стоимость обучения и инференса

Недостатки статистических моделей:

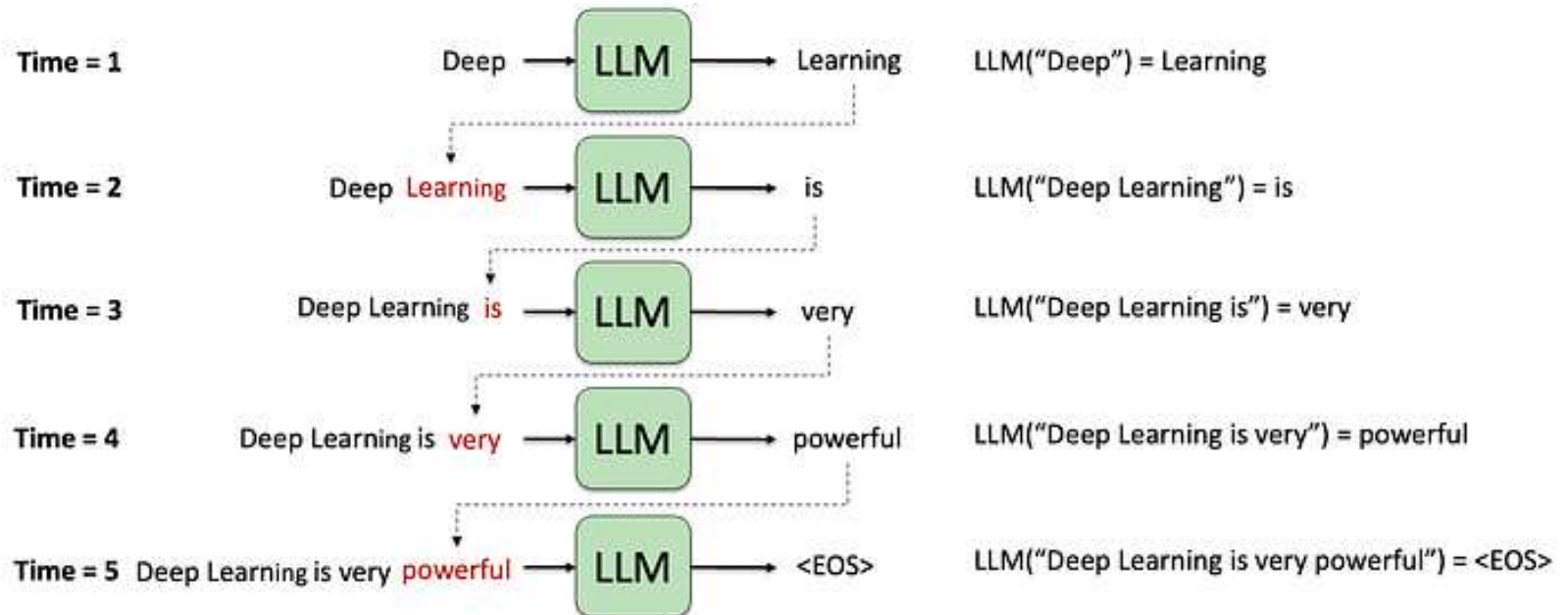
- Не может сгенерировать слова, которые не шли подряд в обучающем корпусе
- Очень маленький контекст
- Длинные последовательности равновероятны и примерно равны нулю, алгоритм не может выдавать разумные продолжения большой длины

Интересно, что такой подход достаточно популярен до сих пор. Например, он используется в умных клавиатурах, чтобы подсказать следующее слово.

Авторегрессионные модели

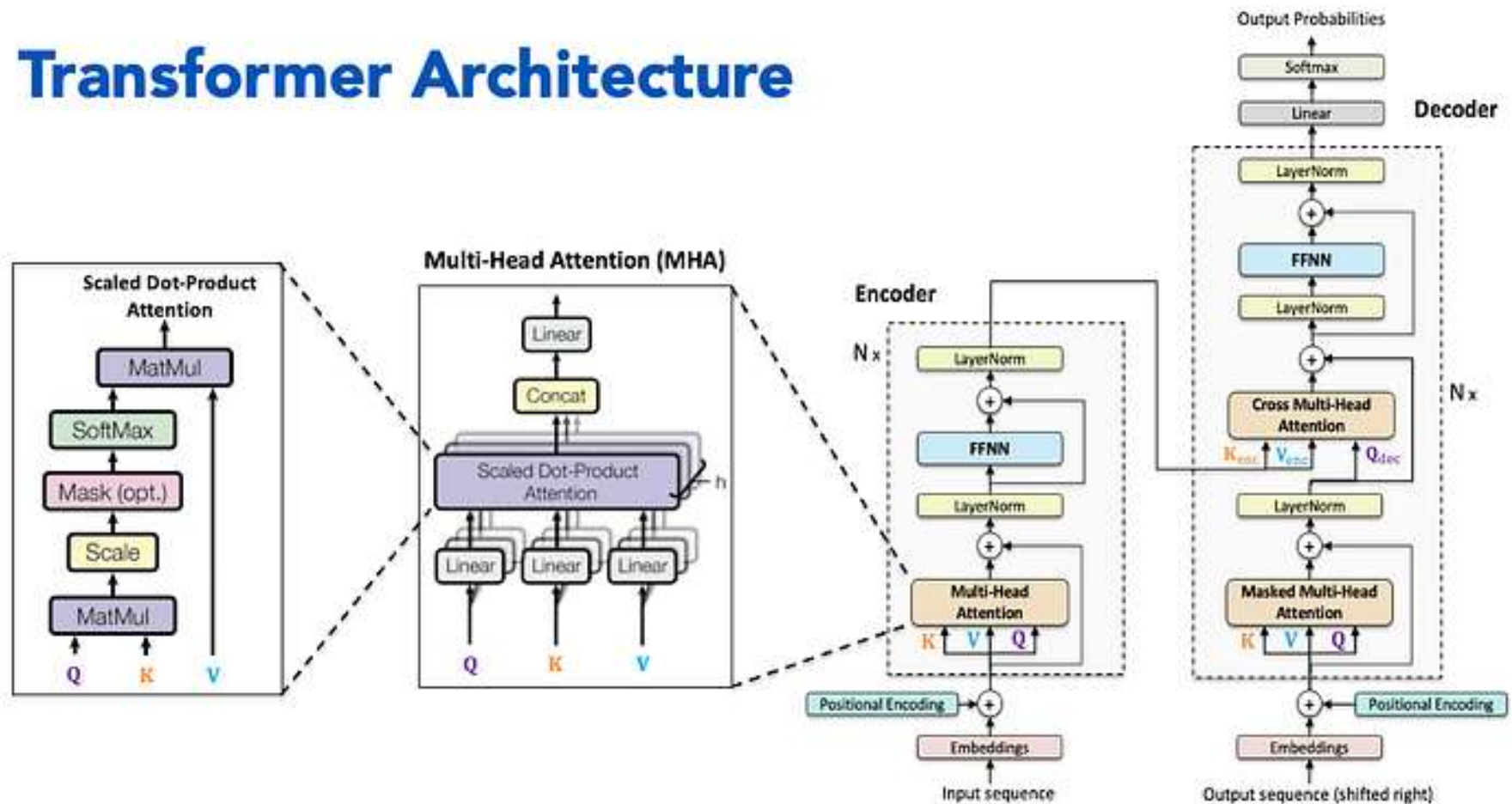
Авторегрессионная природа LLM позволяет им генерировать текст последовательно — по одному токenu за раз, опираясь на контекст, созданный предыдущими словами.

Процесс начинается с начального токена или пользовательского запроса. Затем модель шаг за шагом предсказывает следующий наиболее вероятный токен, продолжая до тех пор, пока не будет сформирована полная последовательность или пока не сработает условие остановки (например, достигнут лимит длины или сгенерирован специальный символ конца). Благодаря такому пошаговому подходу LLM способны создавать связный текст.



Transformers

Transformer Architecture

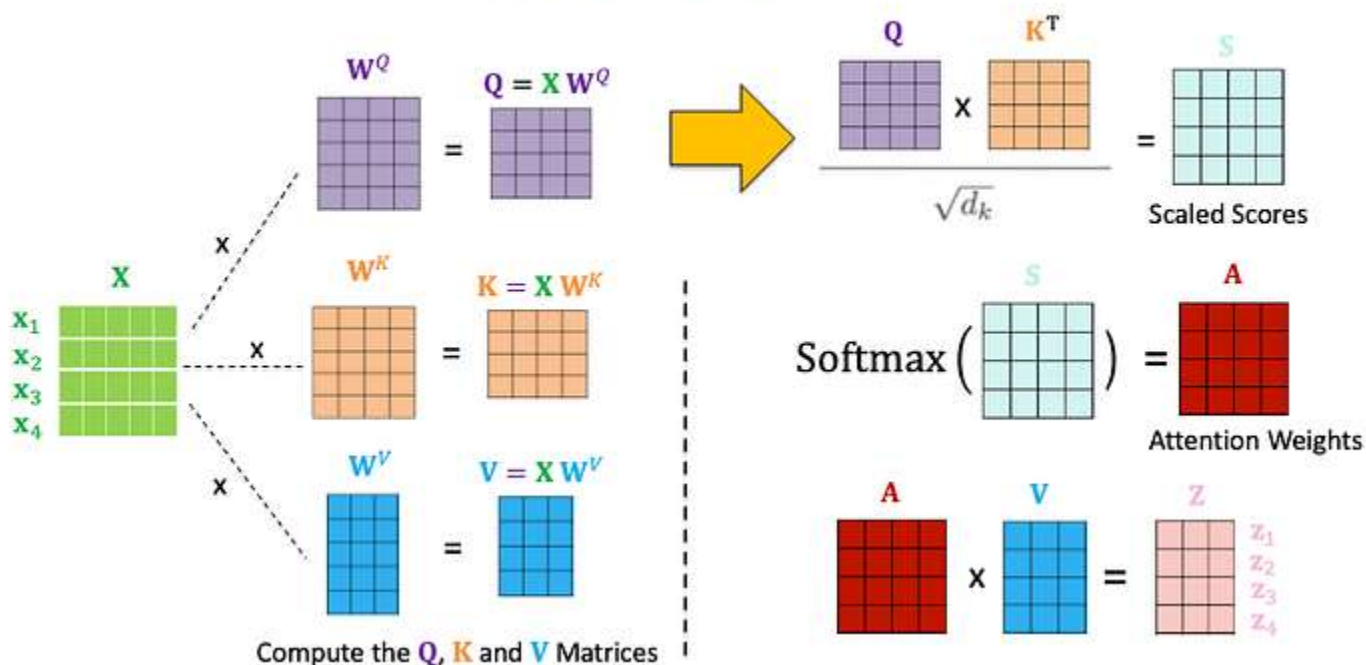


Transformers

Attention is all you need - механизм само-внимания (self-attention) позволяет учитывать все токены в последовательности и на основе сора оценивать на какие части контекста стоит обратить внимание при обработке каждого токена.

$$\text{self-attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

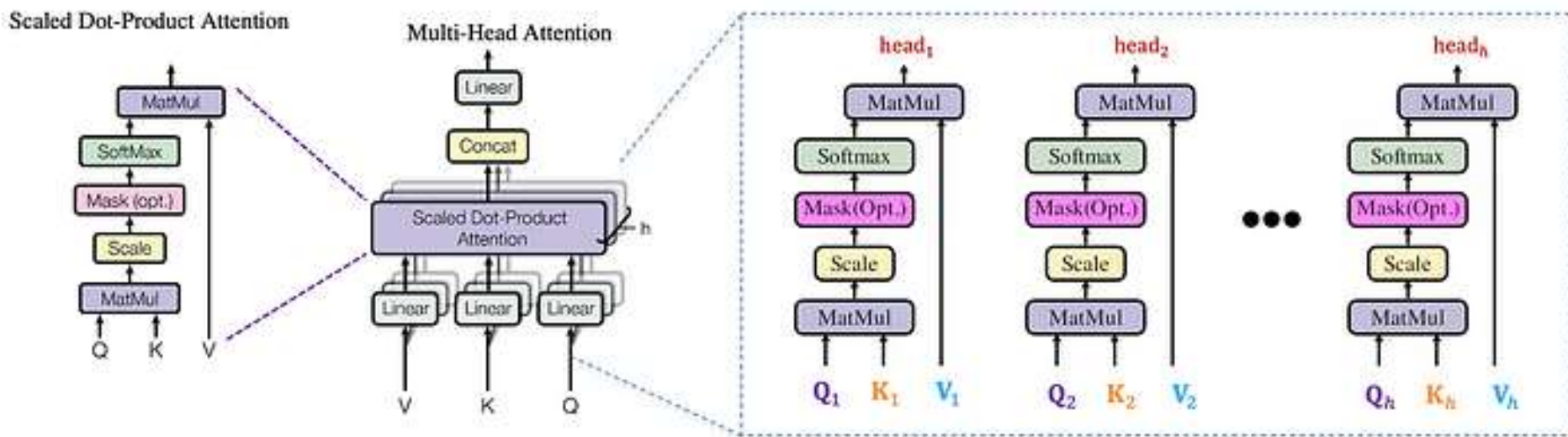
Self-Attention: X, Q, K, V, Z Matrices



Transformers

Multi-Head Attention – обобщение self-attention – параллельное обучение «голов» внимания. Каждая «голова» обучается выделять разные аспекты входной последовательности. Например, одну может интересовать грамматическая роль слов, другую – семантические связи, третью – контекст, четвертую – отношения между сущностями и т.д. Это похоже на множество сверток в слоях CNN.

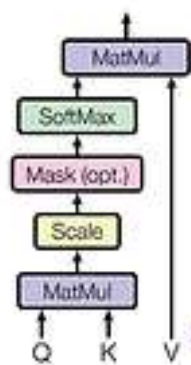
После того, как все головы формируют свои представления, их выходы объединяются (concat) и пропускаются через линейное преобразование для получения итогового векторного представления токена.



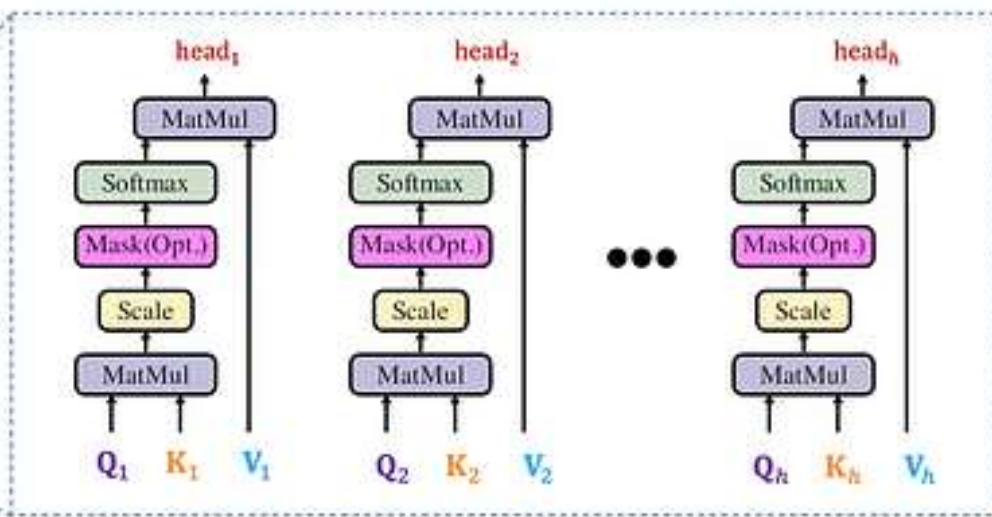
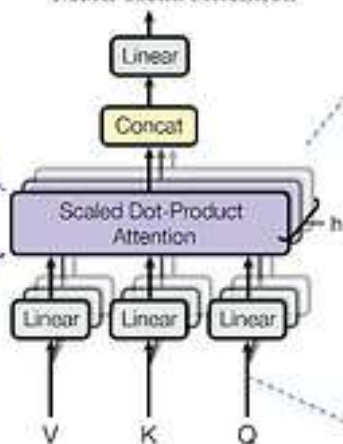
Transformers

FFN + Normalization – два линейных полносвязных слоя (feed forward network), между ними GELU (Gaussian error linear unit) - применяются независимо к каждому токenu

Scaled Dot-Product Attention



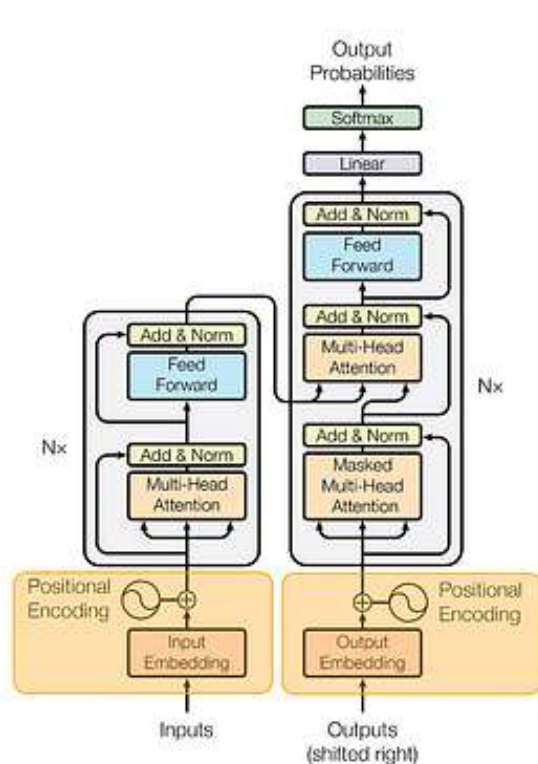
Multi-Head Attention



Transformers

Кодирование позиций (Positional Encoding) – вспомогательные представления (позиционные эмбединги) которые прибавляются к исходным эмбедингам и позволяют слоям внимания различать одинаковые токены на разных местах.

Изначально операции внутри блоков трансформера инвариантны к порядку элементов в последовательности (например attention – скалярное произведение между эмбедингами токенов без учета расположения этих токенов) и на выходе из модели представления каждого токена будут одинаковыми.



Formulation for Positional Encoding

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

$\cos(\frac{e_1 + n}{10000^{2/5}})$	$\sin(\frac{e_2 + n}{10000^{4/5}})$	$\cos(\frac{e_3 + n}{10000^{6/5}})$	$\sin(\frac{e_4 + n}{10000^{8/5}})$	$\cos(\frac{e_5 + n}{10000^{10/5}})$
-------------------------------------	-------------------------------------	-------------------------------------	-------------------------------------	--------------------------------------

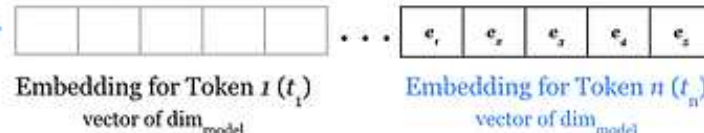
Input Encoding for Token n to the Transformer

© AML.com Research

$\cos(\frac{n}{10000^{2/5}})$	$\sin(\frac{n}{10000^{4/5}})$	$\cos(\frac{n}{10000^{6/5}})$	$\sin(\frac{n}{10000^{8/5}})$	$\cos(\frac{n}{10000^{10/5}})$
$i=1$	$i=2$	$i=3$	$i=4$	$i=5$

Positional Encoding for Token at position n
where, $pos=n$, $d_{model}=5$

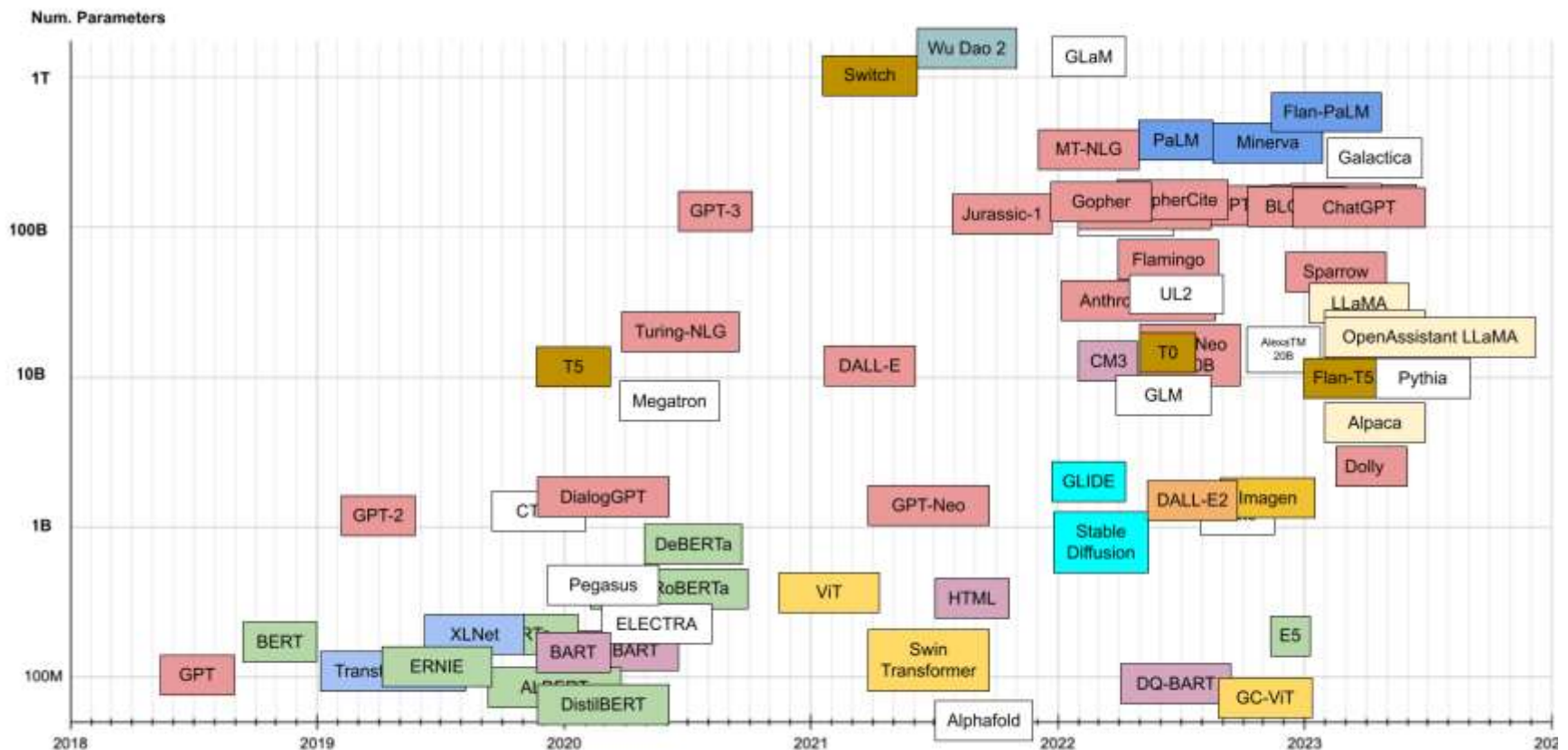
Input Token Embeddings



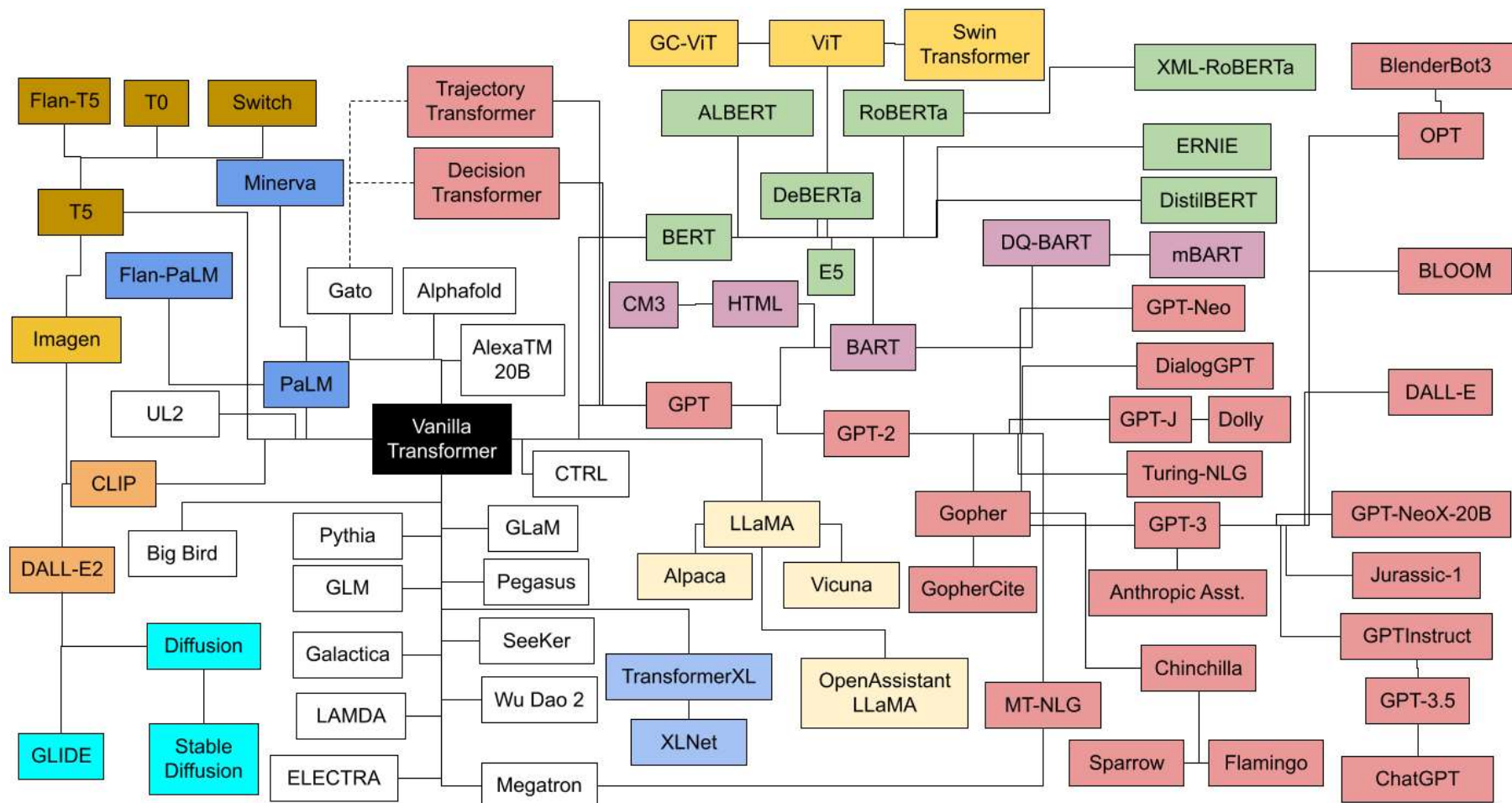
Transformers

Архитектура трансформеров оказала фундаментальное влияние на развитие языковых моделей:

- Scalability (масштабируемость) – благодаря полной параллелизации вычислений архитектура эффективно обучается на огромных датасетах
- Contextual understanding (понимание контекста) – за счет механизма внимания можно одновременно учитывать как локальные, так и глобальные зависимости между словами, даже если они находятся далеко друг от друга в тексте.



Transformers



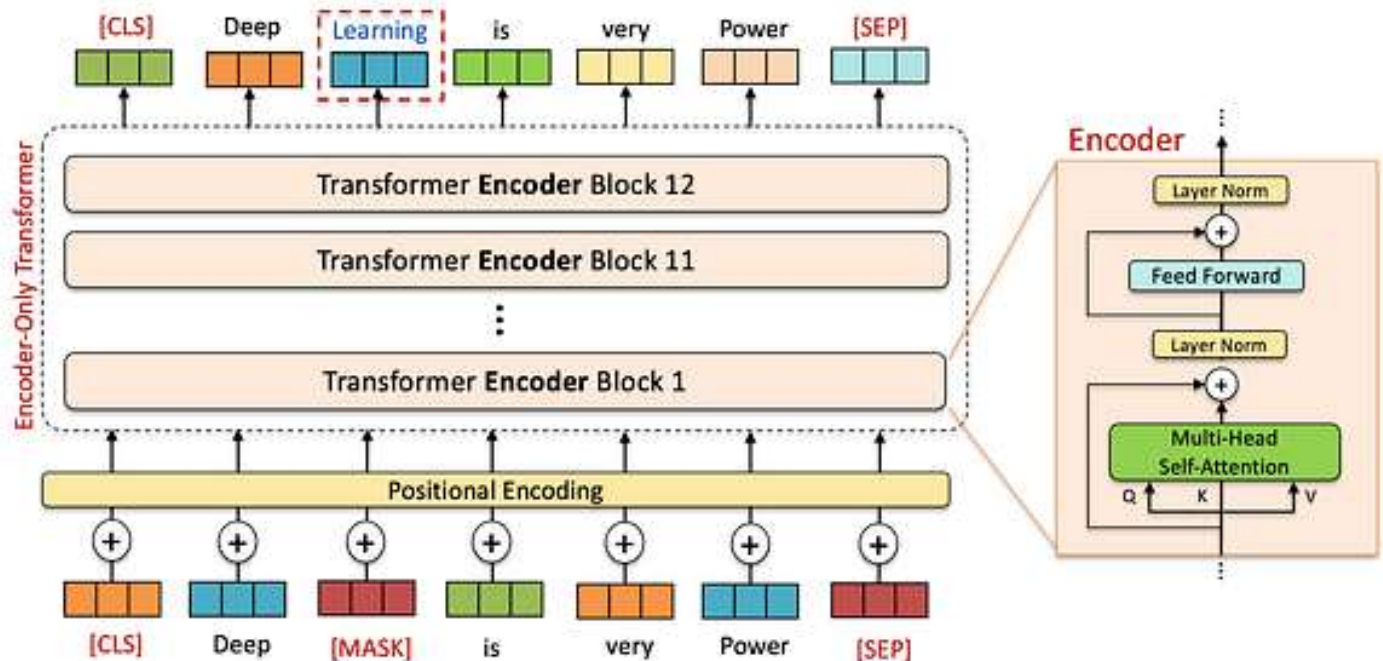
Pre-trained Transformers Era

BERT: Bidirectional Contextual Understanding (2018)

BERT основан на энкодере трансформера и использовал двунаправленный механизм внимания, что позволяло учитывать контекст с обеих сторон и формировать глубокие, контекстуально насыщенные представления для задач понимания языка: классификация текста, распознавание именованных сущностей (NER), анализ тональности и т.д.

BERT не учится генерировать тексты с нуля, он обучается на две задачи:

- Masked language modeling (MLM) – предсказание случайно замаскированных слов по оставшимся
- Next sentence prediction (NSP) – предсказание по паре текстовых фрагментов, следуют они друг за другом или нет



Pre-trained Transformers Era

BERT: Bidirectional Contextual Understanding (2018)

Use the output of the masked word's position to predict the masked word

Possible classes:
All English words

0.1%	Aardvark
...	...
10%	Improvisation
...	...
0%	Zyzzzyva

FFNN + Softmax



Randomly mask
15% of tokens

Input

[CLS] Let's stick to improvisation in this skit

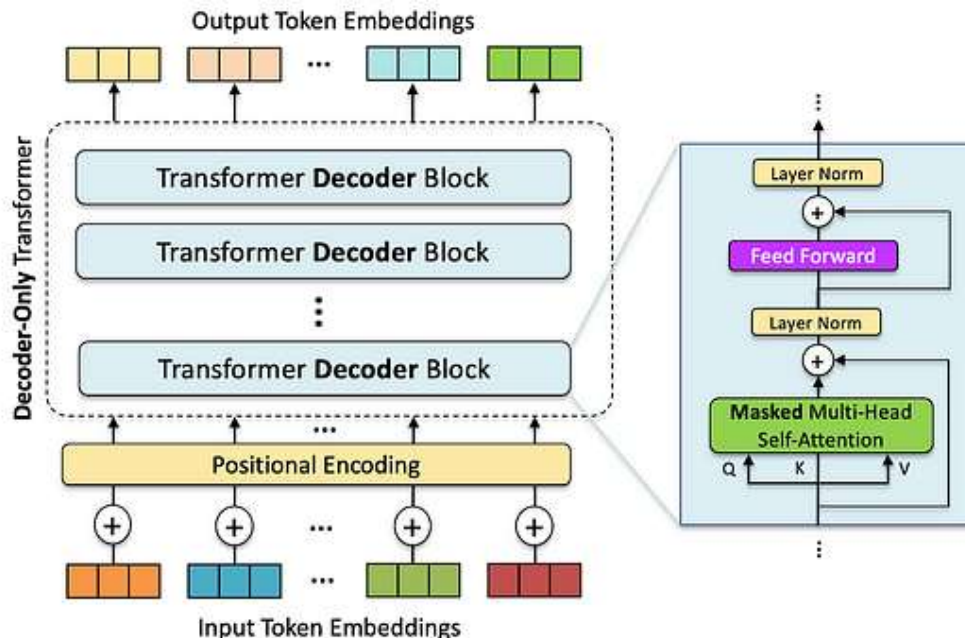
GPT: Generative Pre-trained Transformer (2018-2020)

GPT(2018): unidirectional decoder-only transformer modeling – языковая модель, ориентированная на генеративные способности используя авторегрессионное предобучение. Основана на декодере трансформера и эффективно решает задачу генерации текста.

Первая модель в семействе GPT – 117M параметров.

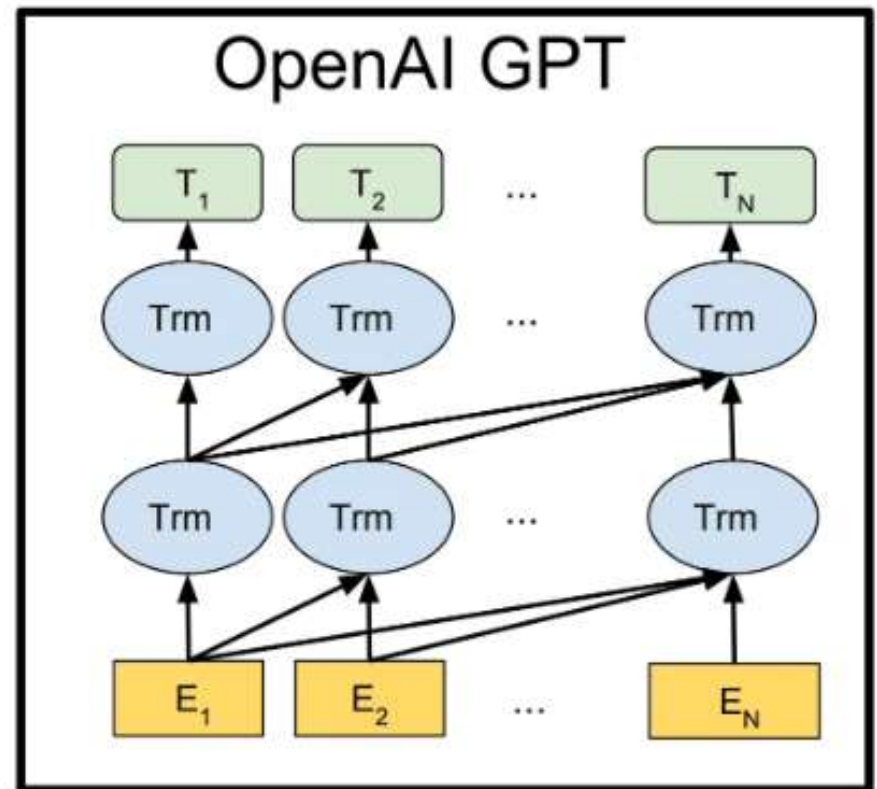
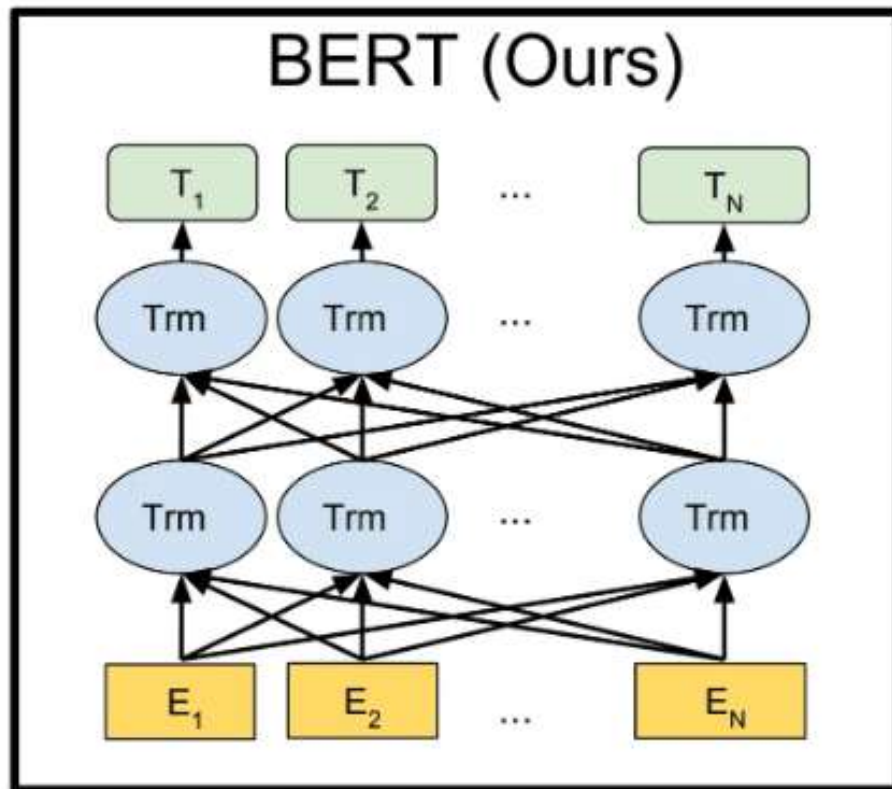
Обучалась на 4.6 ГБ высококачественных текстовых данных используя односторонний (unidirectional) подход: на каждом шаге модель предсказывала следующий токен используя только предыдущие.

Хорошо файнтюнится под конкретные задачи на соответствующих датасетах.



BERT vs GPT

- Ключевое отличие – использование разных видов внимания



GPT-2 (2019): Scaling Up Language Models for Generalized Zero-Shot Learning

GPT-2: масштабирование для универсального обучения без примеров (zero-shot learning)

GPT-2: 1.5B параметров, обучена на 40 ГБ разнообразного текста из интернета.

Главной инновацией модели стали её уникальные способности к zero-shot обучению — то есть выполнению задач без файнтюнинга под конкретные задачи. GPT-2 могла писать сочинения, отвечать на вопросы, составлять краткие пересказы и даже переводить тексты, просто получив соответствующий запрос.

GPT-2 наглядно показала, насколько важны масштаб параметров и объём обучающих данных: чем крупнее модель и данные — тем выше качество генерируемого текста и тем шире спектр «врождённых» способностей.

Zero-shot

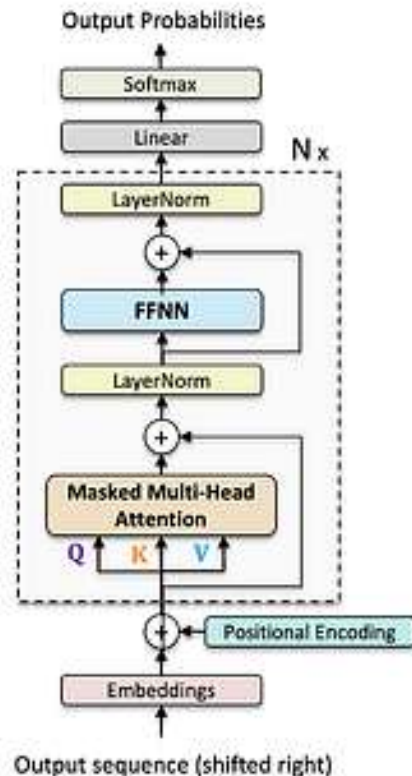
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



GPT-1 (2018) vs GPT-2 (2019)

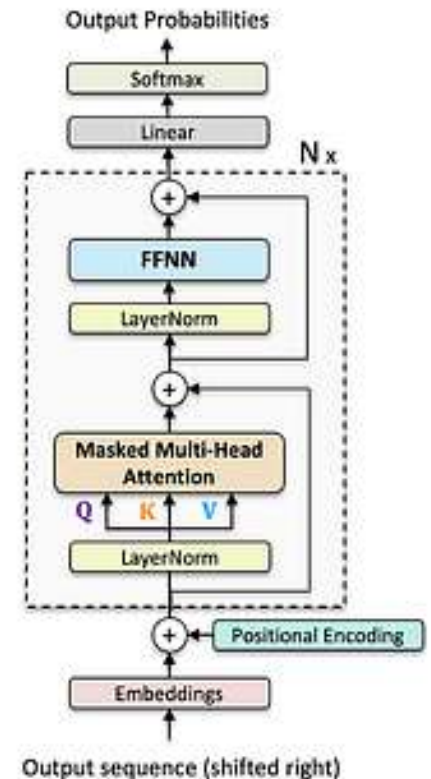
GPT-1 (2018)

- **117M Parameters**
- $N = 12$ Layers
- $h = 12$ heads
- Context window: 256 tokens
- GELU
- **Post-LayerNorm**
- Fine-tune for down-stream NLP tasks



GPT-2 (2019)

- **1.5B Parameters**
- $N = 48$ Layers
- $h = 25$ heads
- Context window: 1024 tokens
- GELU
- **Pre-LayerNorm**
- Strong zero-shot performance



GPT-3 (2020): The Dawn of Massive-Scale LLMs

GPT-3 стала революцией с точки зрения качества и размеров. В 2020 году была получена модель размером в 175 млрд параметров, она обучалась на 570 ГБ текстовых данных с контекстом в 2048 токенов. Модель могла решать целый спектр задач, включая перевод, суммаризацию и ответы на вопросы, с качеством, близким к человеческому уровню, а также отличалась высокой способностью генерировать креативный контент.

Однако для применения таких решений остаётся проблема со стоимостью их обучения. Для обучения GPT-2 авторы использовали 16 GPU (иначе говоря — графических процессоров, видеокарт), а для GPT-3 уже 3200. Для дообучения модели под определенную задачу, конечно, понадобится меньше ресурсов, но всё равно достаточно много. Что с этим делать? Использовать one-shot и few-shot обучение («подводки» - словесное описание поставленной задачи, составленное определенным образом). Это ничего не стоит с точки зрения обучения модели, но показывает хорошие результаты.

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ..... ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```


Post-Training Alignment: Bridging the Gap Between AI and Human Values (2021–2022)

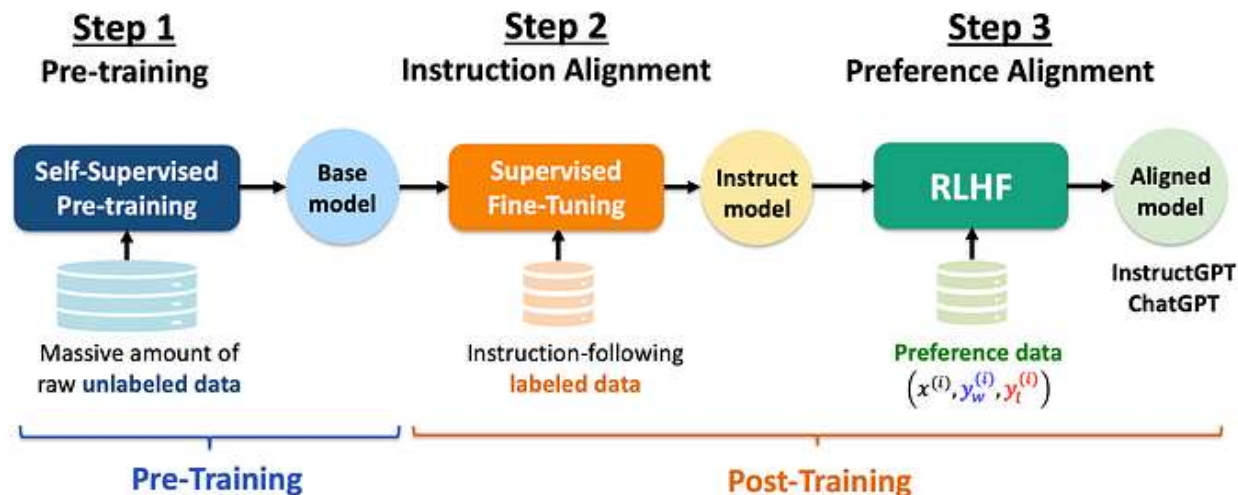
GPT-3 показала что модели способны генерировать текст, очень похожий на человеческий, но оставались две проблемы:

Несмотря на технологический прорыв, стало ясно: сама по себе масштабная предобученная модель не гарантирует соответствие человеческим ценностям, ожиданиям или намерениям пользователя. На практике это проявлялось в двух ключевых проблемах:

- Несогласованность в следовании инструкциям: модель могла игнорировать конкретные указания или уходить от сути запроса.
- Галлюцинации: генерация фактически ложной или бессмысленной информации, выдаваемой за правду.

Для решения этих проблем используется два post-training метода - SFT и RLHF:

1. Supervised fine-tuning (SFT) to enhance Instruction Alignment - модель дообучали на тщательно отобранных наборах данных с примерами корректных ответов на инструкции. Это резко повысило её способность точно следовать указаниям и соблюдать заданные правила.
2. Reinforcement learning from Human Feedback (RLHF) for Preference Alignment – обучение с подкреплением на основе обратной связи от человека – люди оценивали и ранжировали варианты ответов модели и эти оценки использовались для настройки системы вознаграждения. Благодаря этому модель научилась выбирать более предпочтительные, полезные и этичные ответы.



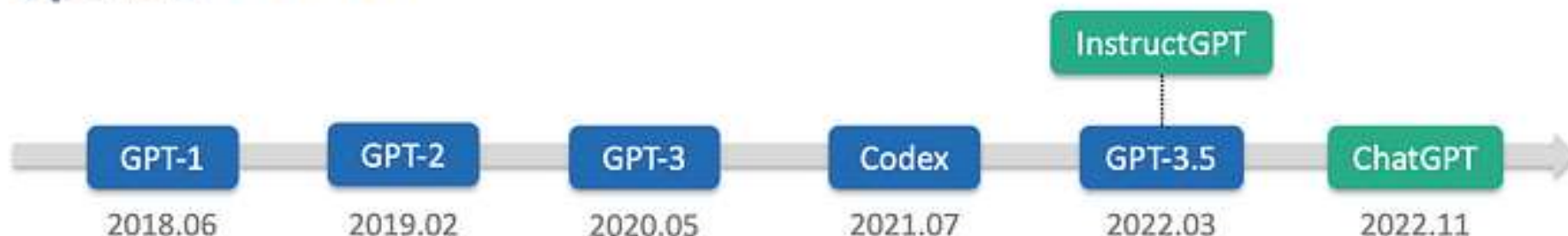
ChatGPT: Advancing Conversational AI (2022)

GPT-3.5 – улучшенная версия GPT-3 с той же архитектурой, но более качественным обучением и файнтюнингом.

InstructGPT – инструкционная модель, обученная отвечать на пользовательские запросы в режиме zero-shot и few-shot с высоким качеством. Обучалась в 4 этапа:

- Претренинг: на качественных данных, которые содержат «все знания мира».
- SFT на инструкционном датасете: инструкция содержит произвольные запросы к модели, а ответ на нее подробный текст, который подходит пользователю.
- Обучение reward-модели: Каждый ответ алгоритма можно оценить с точки зрения вежливости, подробности или персонажности. Персонажность позволяет модели считать себя, например, капитаном Джеком Воробьем и общаться на пиратском говоре.
- RL: модель обучается генерировать такие ответы, которые имеют максимальный reward

OpenAI GPT



ChatGPT: Advancing Conversational AI (2022)

ChatGPT: наследует логику обучения от InstructGPT, но отличается от нее:

- Диалоговость: модель обучена работать с диалогами, держит их в контексте и помнит историю того, что требовал пользователь. Обучение производится путем сбора и написания диалоговых данных
- Увеличили размер и улучшили качество инструкционного датасета
- Улучшена разметка и обучение reward-модели и этапу RL

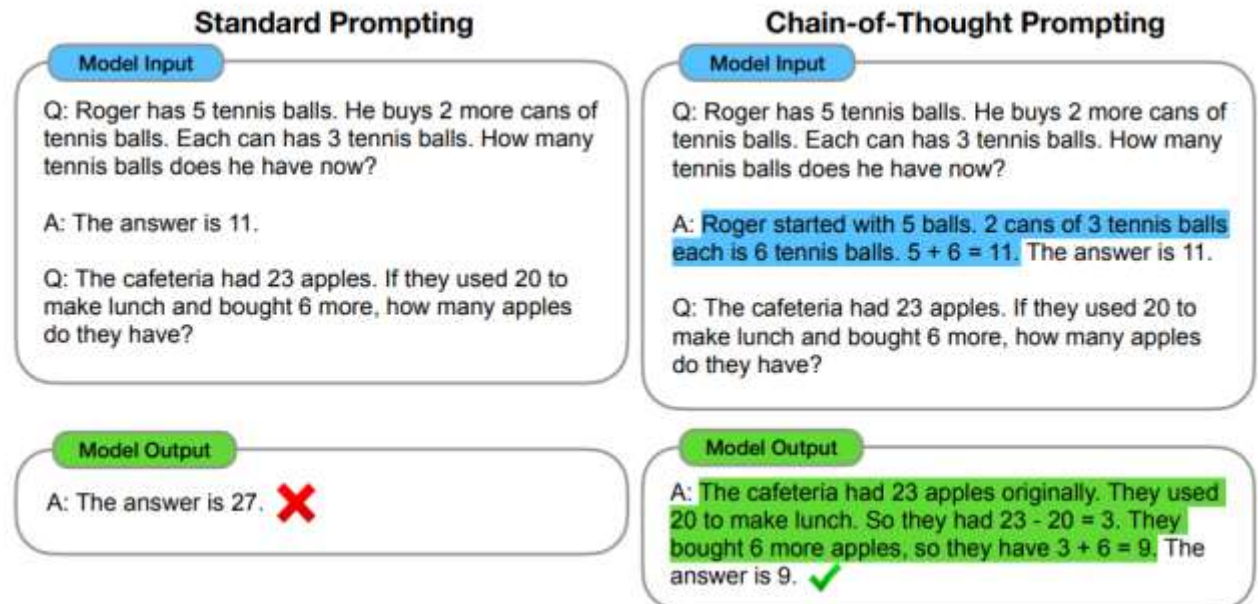
Запуск ChatGPT стал поворотным моментом в истории языковых моделей (ChatGPT moment). Эта модель наглядно продемонстрировала, как диалоговые модели кардинально меняют способ взаимодействия человека и машины, сделав его интуитивным, доступным и массовым.

Что еще придумали?

Chain-of-Thought (CoT) – для ответов на вопросы, которые требуют рассуждения, которое модель не может просто так провести из-за своей архитектуры.

CoT состоит из трех элементов:

- Формулировка задачи на естественном языке
- Подробное пошаговое решение
- Ответ на задачу



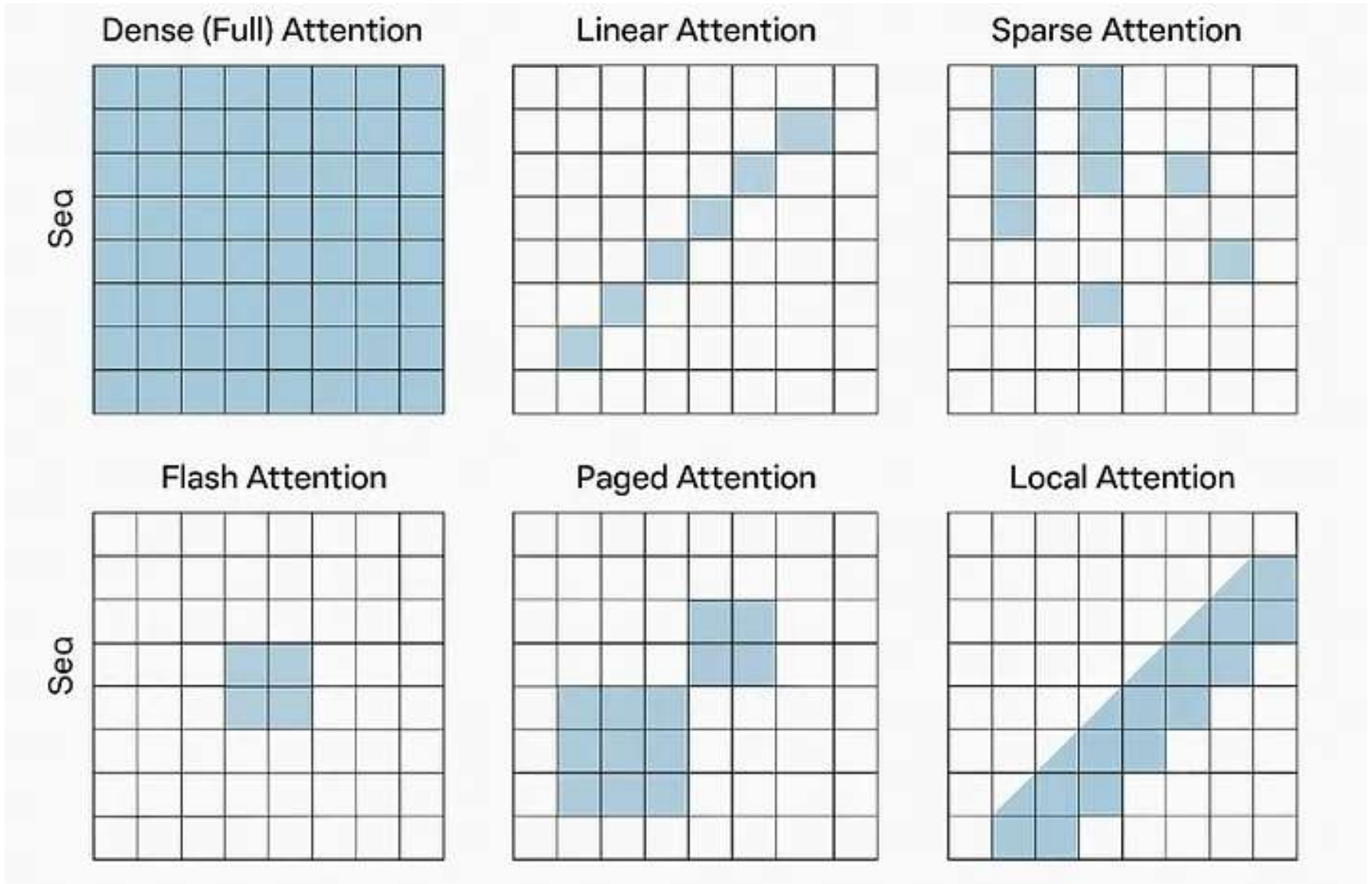
Что еще придумали?

The Evolution of Attention Mechanisms: Scaling Transformers Smartly

1. Dense (Full) Attention – каждый токен взаимодействует со всеми остальными, дает максимальный контекст но квадратичная сложность
2. Linear Attention – заменяет полную матрицу внимания на приближенные вычисления, позволяет снижать сложность до линейной
3. Sparse Attention – каждый токен взаимодействует лишь с небольшим набором токенов (через маски)
4. Flash Attention – оптимизация вычисления Dense Attention на уровне железа (минимизация операций обмена данными между GPU памятью и кэшем)
5. Paged Attention – не загружает весь контекст целиком, используя контролируемые по памяти «страницы»
6. Local Attention – каждый токен смотрит только на соседей в окне фиксированного размера

Что еще придумали?

The Evolution of Attention Mechanisms: Scaling Transformers Smartly



Multimodal Models: Bridging Text, Images, and Beyond (2023–2024)

Open-Source and Open-Weight Models (2023–2024)

GPT-4V: Vision Meets Language

GPT-4o: The Omni-Modal Frontier

Prompt:

Describir la imagen en 20 idiomas diferentes



GPT-4V:

1. Inglés: This is a beautiful view of Machu Picchu in Peru.
2. Español: Esta es una hermosa vista de Machu Picchu en Perú.
3. Francés: C'est une belle vue de Machu Picchu au Pérou.

GPT-4V output continued

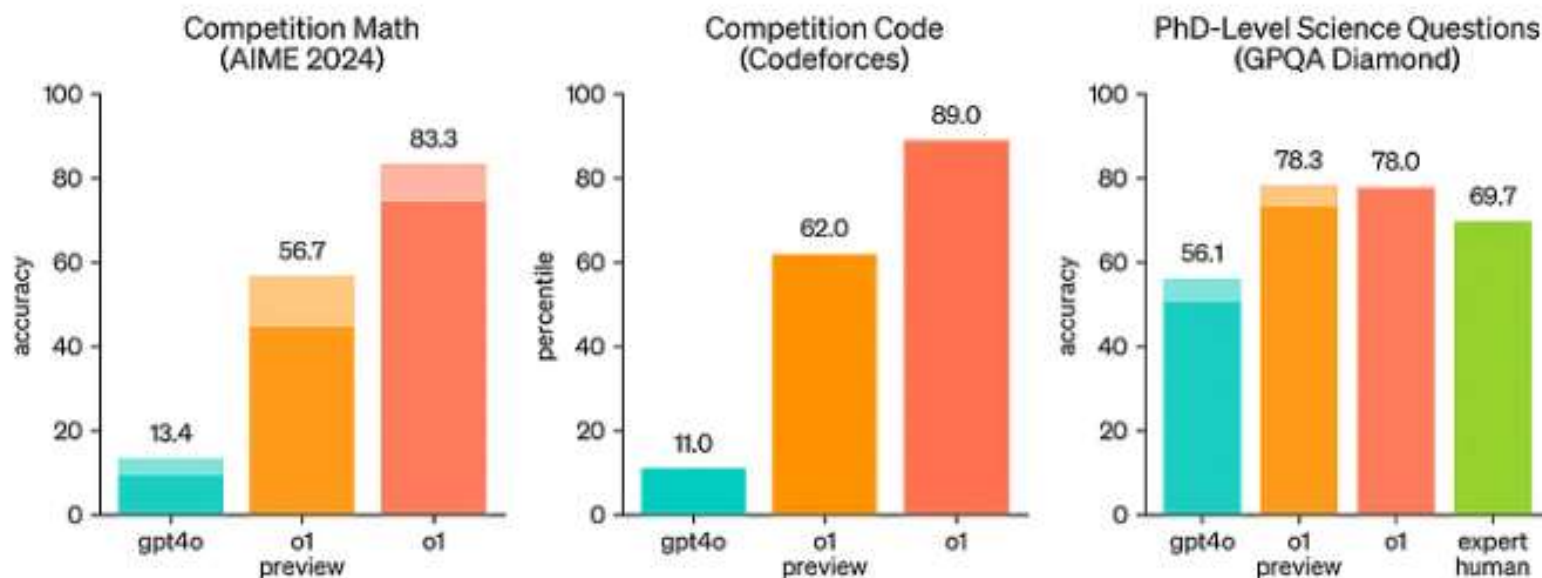
4. Alemán: Dies ist eine schöne Aussicht auf Machu Picchu in Peru.
5. Italiano: Questa è una bella vista di Machu Picchu in Perù.
6. Portugués: Esta é uma bela vista de Machu Picchu no Peru.
7. Ruso: Это красивый вид на Мачу-Пикчу в Перу.
8. Chino: 这是秘鲁马丘比丘的美丽景色。
9. Japonés: これはペルーのマチュピチュの美しい景色です。
10. Coreano: 이것은 페루의 마추 피추의 아름다운 경치입니다.
11. Hindi: यह पेरू के माचू पिचू का सुंदर दृश्य है।
12. Árabe: هذا منظر جميل لمأثو بيتشو في بيرو.
13. Turco: Bu, Peru'daki Machu Picchu'nun güzel bir manzarasıdır.
14. Griego: Αυτή είναι μια όμορφη θέα του Μάτσου Πίτσου στο Περού.
15. Hebreo: זהו נוף יפה של מאצ'ו פיצ'ו בפרו.
16. Sueco: Det här är en vacker vy av Machu Picchu i Peru.
17. Noruego: Dette er en vakker utsikt over Machu Picchu i Peru.
18. Danés: Dette er en smuk udsigt over Machu Picchu i Peru.
19. Holandés: Dit is een prachtig uitzicht op Machu Picchu in Peru.
20. Polaco: To piękny widok na Machu Picchu w Peru.

Reasoning Models: The Shift from System 1 to System 2 Thinking (2024)

Модели со способностью к рассуждению: OpenAI-o1, o3:

- Длинные CoT: модель разбивает задачу на шаги, оценивает промежуточные решения, пробует различные стратегии – похоже на «мыслительный процесс»
- Inference-time compute control: оптимизация инференса, эффективные вычисления

Как следствие – улучшения на множестве бенчмарков для оценки моделей.



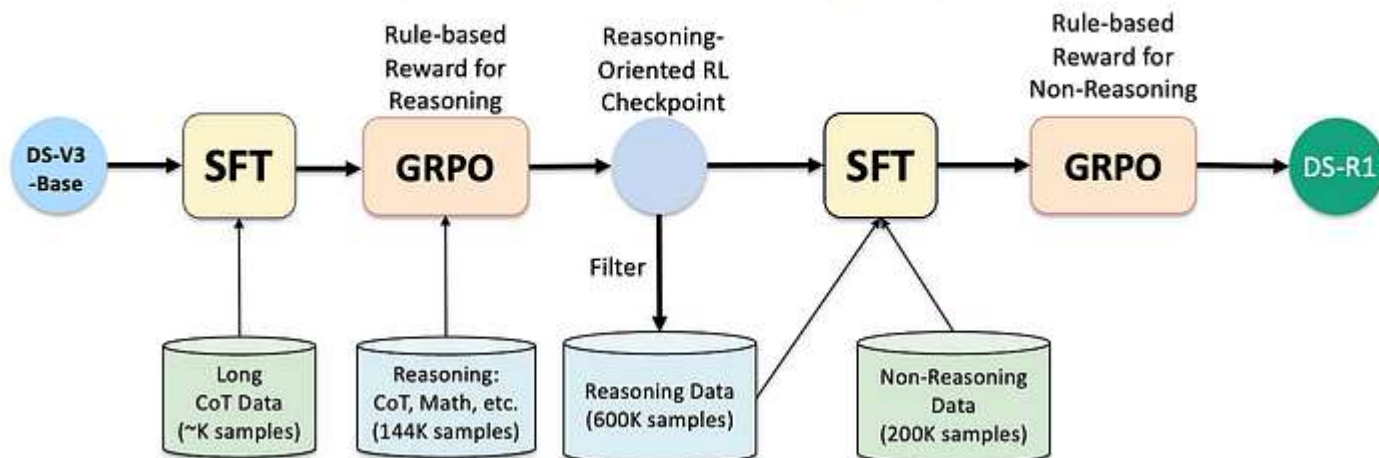
GPT-4o vs o1-preview vs o1 vs expert human.

Cost-Efficient Reasoning Models: DeepSeek-R1 (2025)

DeepSeek-R1 – полностью открытая модель с тремя ключевыми особенностями:

- Multi-Head Latent Attention (MLA) - усовершенствованная версия механизма внимания, обеспечивающая более глубокое и компактное представление контекста
- DeepSeek Mixture of Experts (DeepSeekMoE) - механизм, при котором для каждого запроса активируются только наиболее релевантные «экспертные» подсети, что резко снижает вычислительные затраты без потери качества
- Multi-Token Prediction - способность предсказывать сразу несколько токенов за один шаг, ускоряя генерацию и повышая согласованность вывода.

DeepSeek-R1 Training Pipeline



AI Leaderboards

Top models ranked by performance

LLM

Image Generation

Video Generation































Text-to-Speech

Speech-to-Text

Embeddings

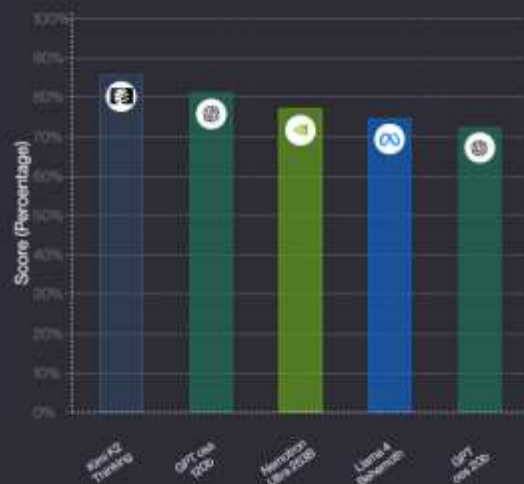
View full leaderboard →

199 MODELS

RANK	MODEL	GPQA ↓	AIME 2025	SWE-BENCH	CODE ARENA	CONTEXT	INPUT \$/M	OUTPUT \$/M	LICENSE
1	 Gemini 3 Pro (Google)	91.9%	100.0%	76.2%	1 867	1.0M	\$2.00	\$12.00	
2	 Grok-4 Heavy (xAI)	88.4%	100.0%	—	—	—	—	—	
3	 GPT-5.1 Instant (OpenAI)	88.1%	94.0%	76.3%	126	400K	\$1.25	\$10.00	
4	 GPT-5.1 Thinking (OpenAI)	88.1%	94.0%	76.3%	24	400K	\$1.25	\$10.00	
5	 GPT-5.1 (OpenAI)	88.1%	94.0%	76.3%	—	400K	\$1.25	\$10.00	
6	 Grok-4 (xAI)	87.5%	91.7%	—	-389	256K	\$3.00	\$15.00	
7	 Claude Opus 4.5 (Anthropic) NEW	87.0%	—	80.9%	1 463	200K	\$5.00	\$25.00	
8	 Gemini 2.5 Pro Preview 06-05 (Google)	86.4%	88.0%	67.2%	—	1.0M	\$1.25	\$10.00	
9	 GPT-5 (OpenAI)	85.7%	94.6%	74.9%	-766	400K	\$1.25	\$10.00	
10	 Grok 4 Fast (xAI)	85.7%	92.0%	—	—	2.0M	\$0.20	\$0.50	
11	 Claude 3.7 Sonnet (Anthropic)	84.8%	54.8%	70.3%	-226	200K	\$3.00	\$15.00	
12	 Grok-3 (xAI)	84.6%	93.3%	—	—	128K	\$3.00	\$15.00	
13	 Grok-3 Mini (xAI)	84.0%	90.8%	—	-816	128K	\$0.30	\$0.50	
14	 Claude Sonnet 4.5 (Anthropic)	83.4%	87.0%	—	-353	200K	\$3.00	\$15.00	
15	 o3 (OpenAI)	83.3%	86.4%	69.1%	—	200K	\$2.00	\$8.00	

Best open source models per task

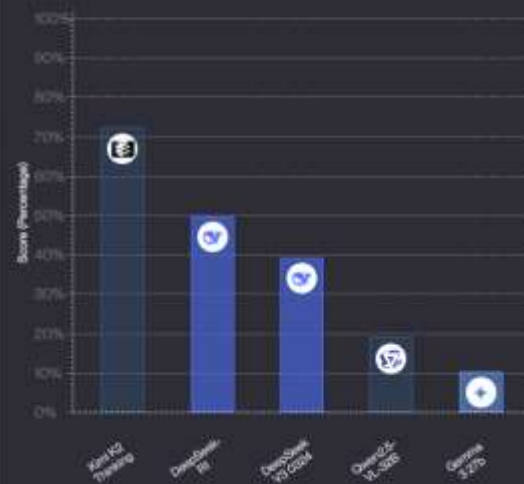
Best in Reasoning (GPQA Diamond) ⓘ



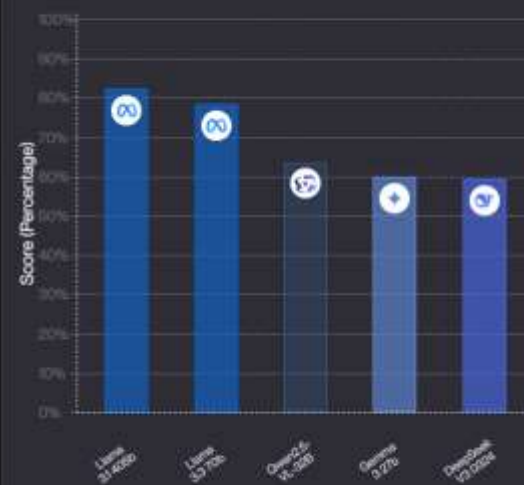
Best in High School Math (AIME 2024) ⓘ



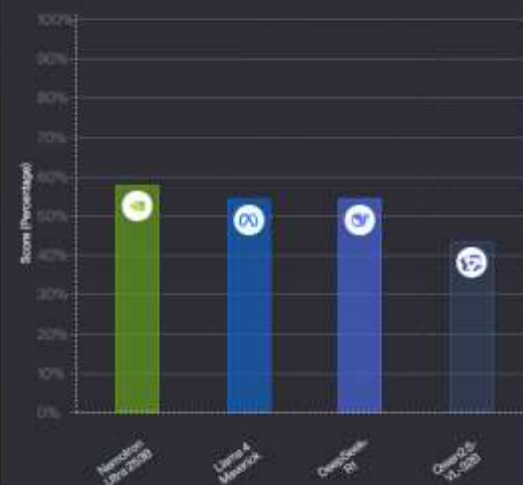
Best in Agentic Coding (SWE Bench) ⓘ



Best in Tool Use (BFCL) ⓘ



Best in Adaptive Reasoning (GRIND) ⓘ



Best Coding (LiveCode Bench) ⓘ

