

Лекция 8. Предобученные Трансформеры

Денис Деркач, Дмитрий Тарасов

Использовались слайды Антона Кленицкого, Лены Войта

24 марта 2025 года



В предыдущих лекциях

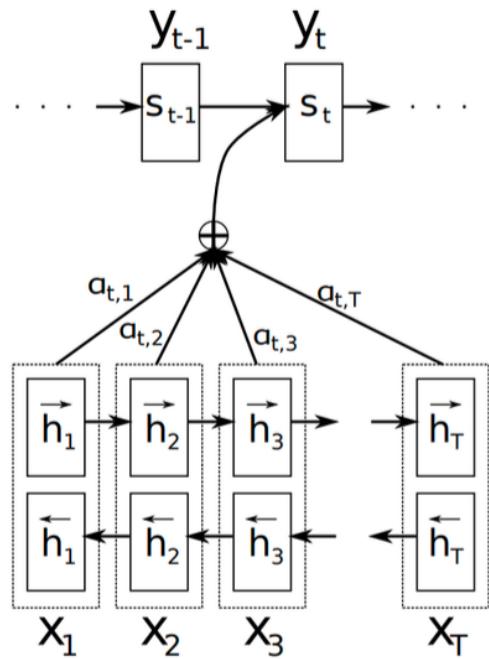


Методы построения эмбедингов

- ▶ Word2Vec: используйте, когда семантические связи имеют решающее значение, и у вас большой набор данных.
- ▶ GloVe: подходит для разнообразных наборов данных и когда важен охват глобального контекста.
- ▶ FastText: выбирайте морфологически богатые языки или когда обработка слов, не входящих в словарный запас, имеет решающее значение.

Attention

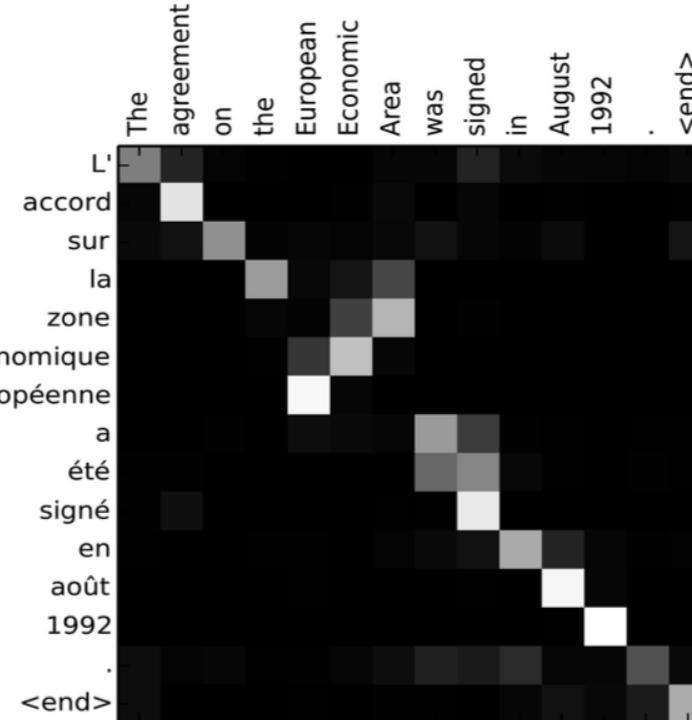
Bahdanau, D., Cho, K., & Bengio, Y. (2015) Neural machine translation by jointly learning to align and translate.



- Умный pooling
- Вектор контекста c_t свой на каждом шаге декодера
- c_t - сумма h_i со всех шагов энкодера с обучаемыми весами
- Внимание выбирает релевантные элементы во входной последовательности

Интерпретация внимания

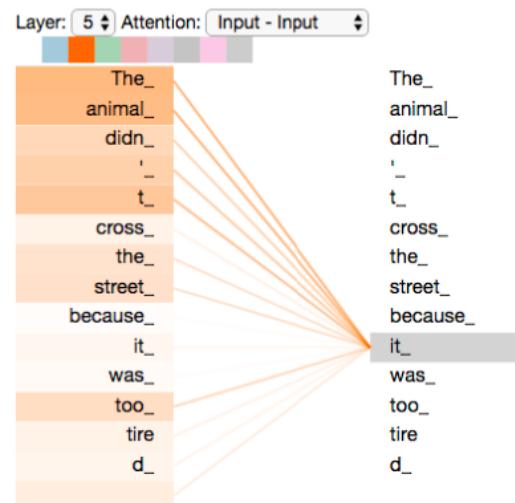
Матрица весов α_{ti} определяет alignment (выравнивание) элементов входной и выходной последовательностей



<https://arxiv.org/pdf/1409.0473>

Self-attention

- Attention позволяет получить представление входной последовательности
- как и RNN
- Почему бы не отказаться от рекуррентности полностью?



Self-attention

x_i - элементы последовательности с размерностью d

$$q_i = W_q x_i \quad \text{query}$$

$$k_i = W_k x_i \quad \text{key}$$

$$v_i = W_v x_i \quad \text{value}$$

Добавили матрицы $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ для большей выразительности

$$y_i = \sum_j \text{softmax}\left(\frac{q_i^T k_j}{\sqrt{d}}\right) v_j$$

Делим на \sqrt{d} , чтобы лучше обучалось, градиенты софтмакса не были слишком маленькими

Матричный вид Self-attention

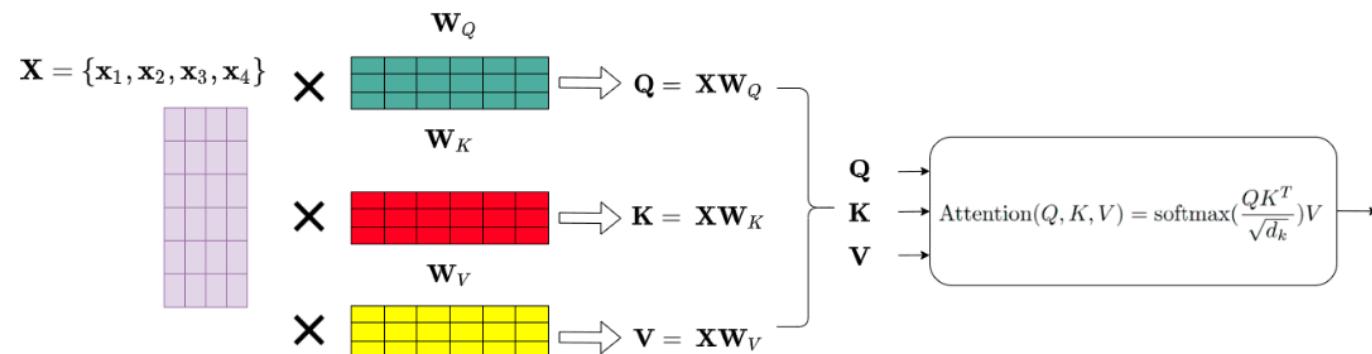
В матричном виде

$$Q = XW^Q$$

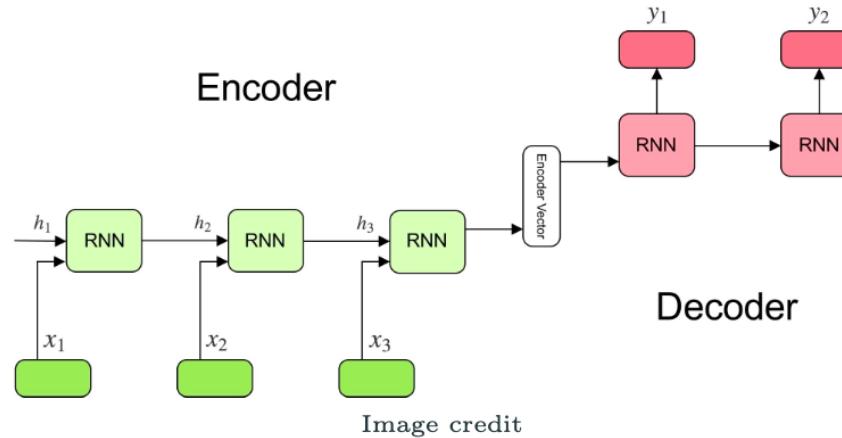
$$K = XW^K$$

$$V = XW^V$$

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d}} \right) V$$



Архитектура кодировщик-декодировщик

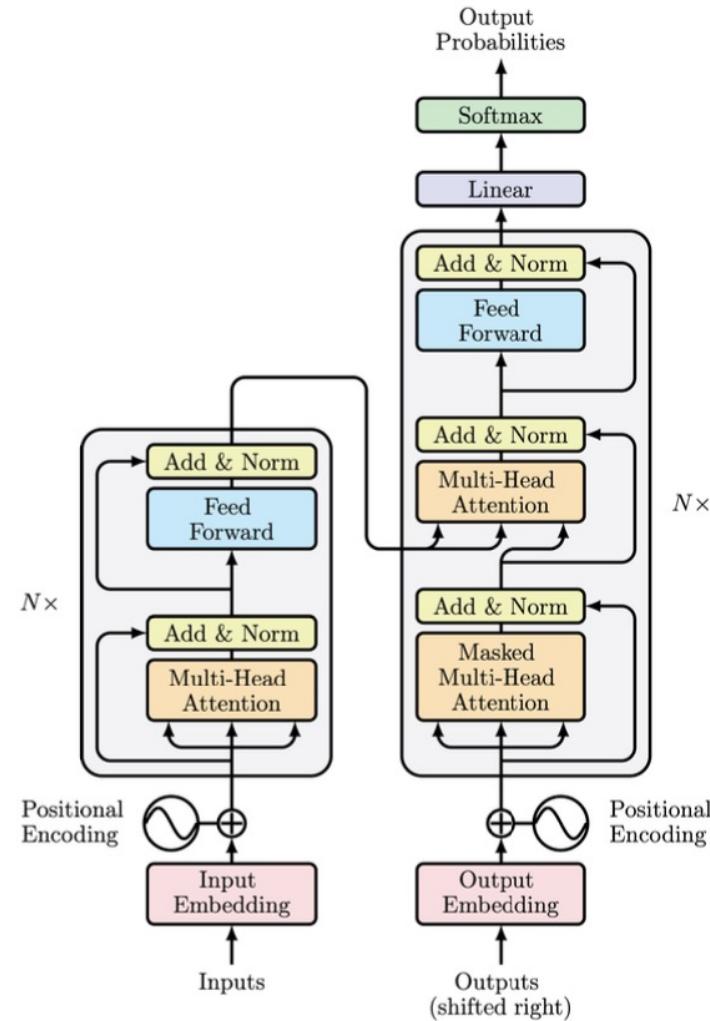


Encoder сворачивает входную последовательность в вектор контекста c (как правило, последнее скрытое состояние h_T):

Decoder предсказывает следующий элемент на основе скрытого состояния s_t , предыдущего элемента y_{t-1} и вектора контекста c

Архитектура

- Vaswani A. et al. (2017)
Attention is all you need.
- Encoder-Decoder
архитектура
- Изначально придумали
для machine translation
- Трансформеры стали
универсальной
архитектурой для
обработки
последовательностей



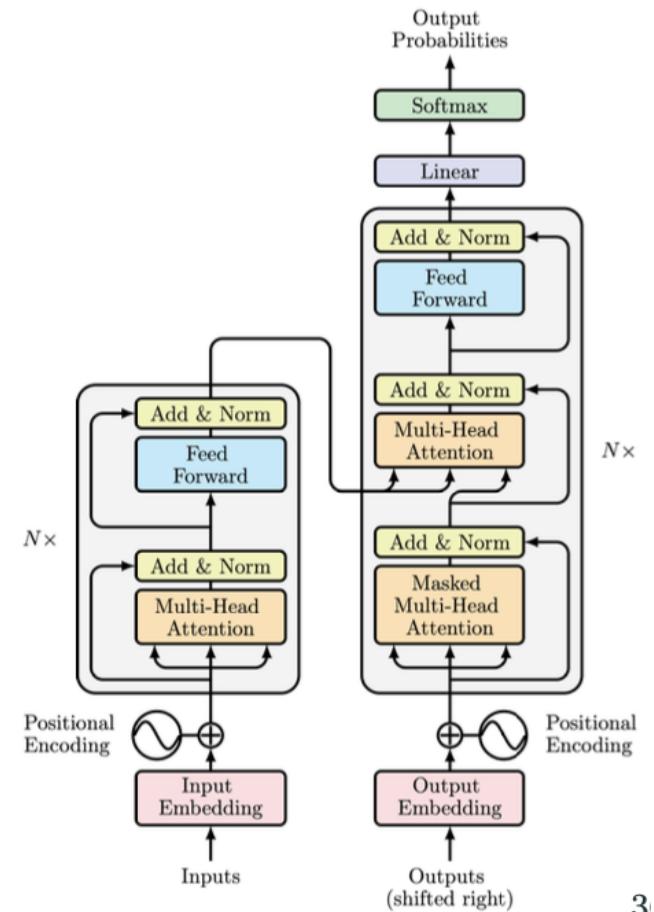
Трансформер

Pros

- Self-attention может улавливать долгосрочные зависимости
- Параллелизуется, в отличие от RNN

Cons

- Квадратичная зависимость self-attention от длины последовательности



3

BERT-like модели для текстов



Перенос обучения (transfer learning)

Центральная идея современного NLP

- ▶ **Pretrain** - предобучаемся на общей задаче с большим количеством данных и дешевой (лучше совсем бесплатной) разметкой
- ▶ **Finetune** - дообучаем модель под конкретную задачу

NLP's ImageNet moment - 2018

Разное предобучение

Encoder-only

- Классификация
- ELMo, BERT, RoBERTa

Decoder-only

- Text generation
- GPT family

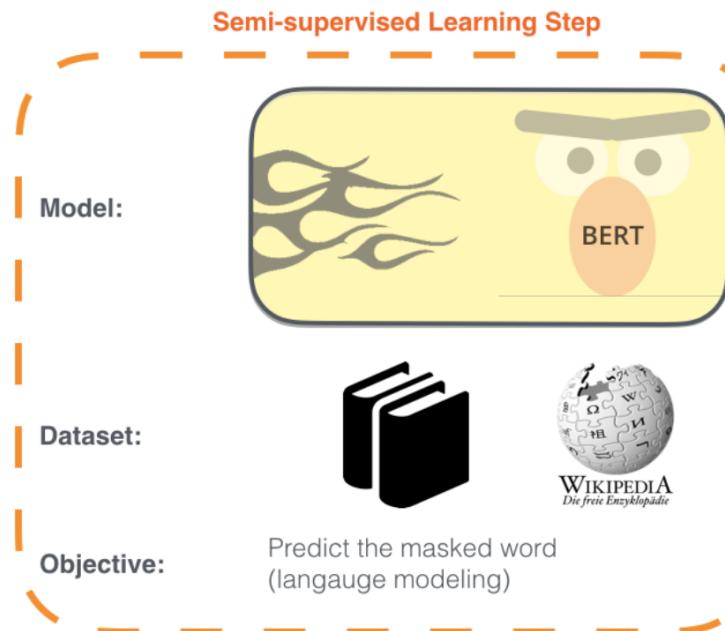
Encoder-Decoder

- Translation, summarization
- BART, T5

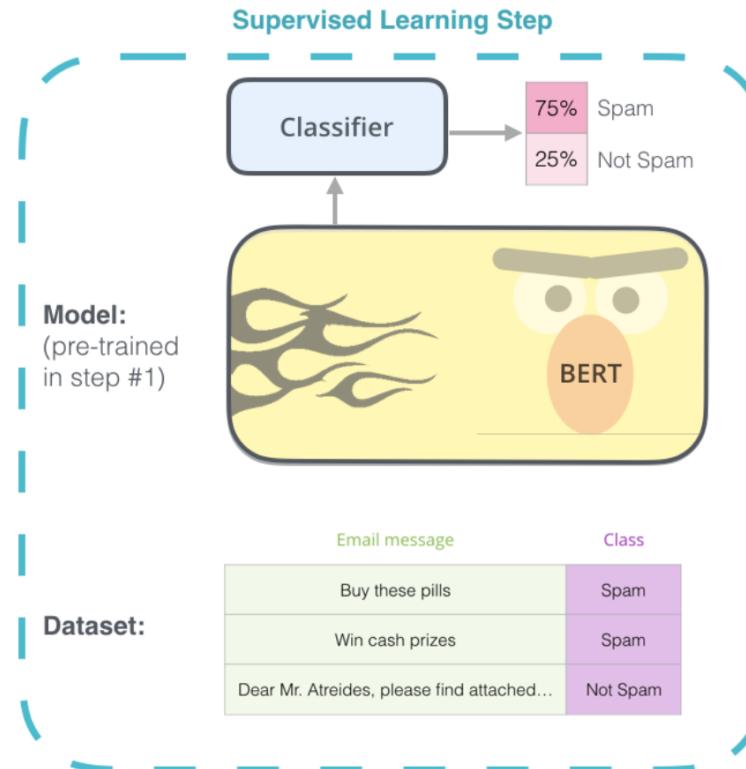
Transfer Learning

1 - **Semi-supervised** training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.



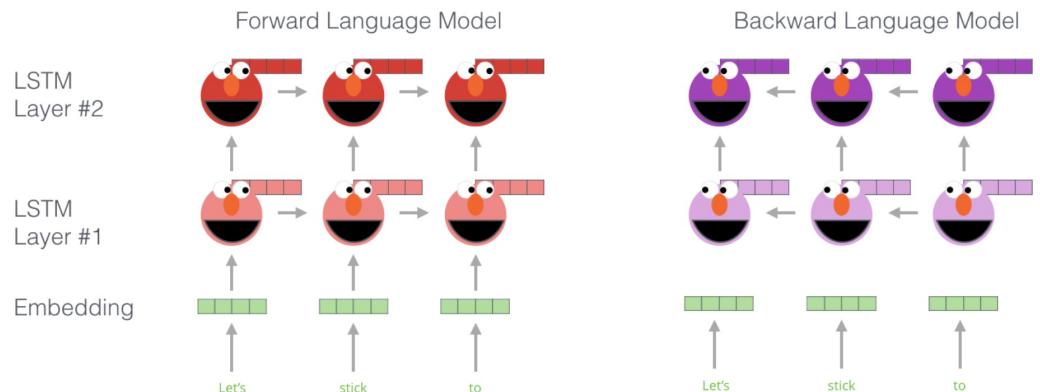
2 - **Supervised** training on a specific task with a labeled dataset.



Embeddings from Language Model (ELMo)

- ▶ Контекстно зависимые представления (в отличие от word2vec)
- ▶ Представления слов - свертки поверх представлений символов
- ▶ 2-layer LSTM в прямом и обратном направлении

Embedding of "stick" in "Let's stick to" - Step #1

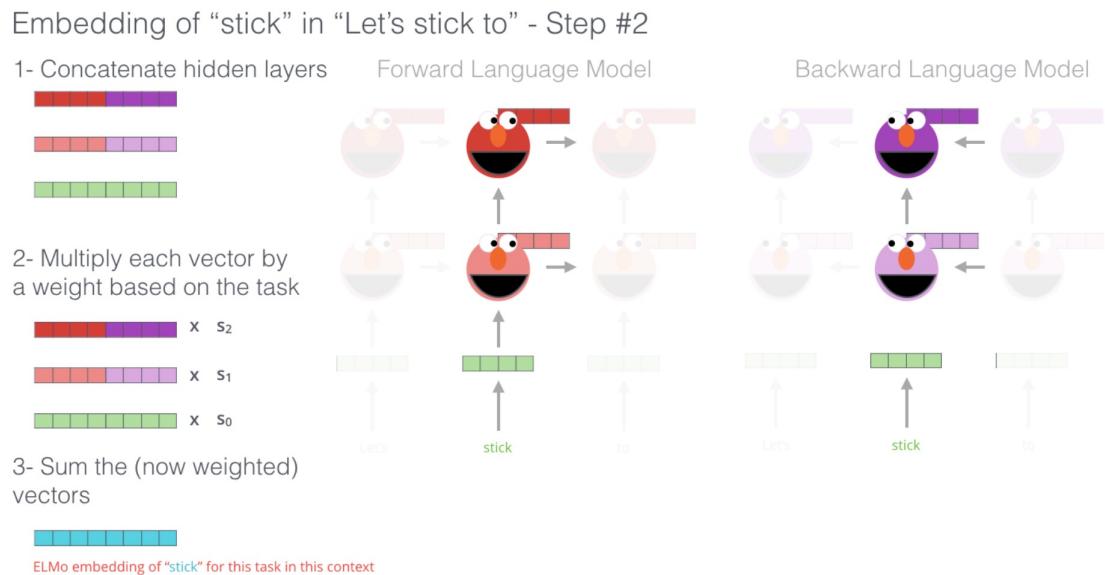


<https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/>

Peters M. et al. (2018) Deep contextualized word representations. <https://arxiv.org/abs/1802.05365>

Embeddings from Language Model (ELMo)

- ▶ Контекстно зависимые представления (в отличие от word2vec)
- ▶ Представления слов - свертки поверх представлений символов
- ▶ 2-layer LSTM в прямом и обратном направлении



TASK	PREVIOUS SOTA	OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17
SRL	He et al. (2017)	81.7	81.4	84.6
Coref	Lee et al. (2017)	67.2	67.2	70.4
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5

<https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/>

Peters M. et al. (2018) Deep contextualized word representations. <https://arxiv.org/abs/1802.05365>

Bidirectional Encoder Representations from Transformers (BERT)

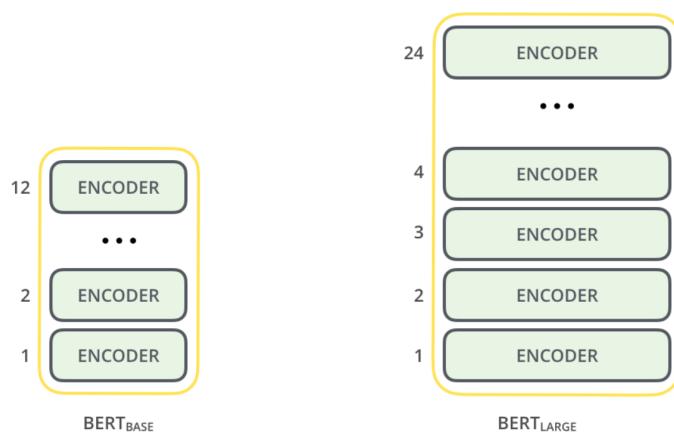
Transformer Encoder

- ▶ BERT base L=12, H=768, A=12, 110M параметров
- ▶ BERT large L=24, H=1024, A=16, 340M параметров

L - количество слоев

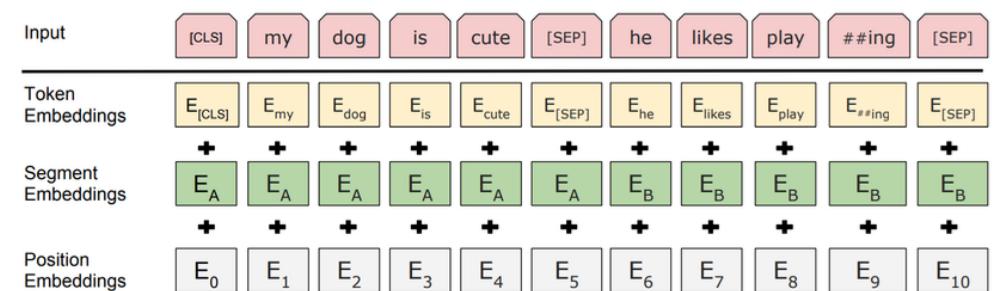
H - размерность скрытого состояния

A - количество голов внимания



Представление входных данных

- Subword tokenization
- Специальные токены: [CLS], [SEP]
- Token + Position + Segment embeddings



Bidirectional Encoder Representations from Transformers (BERT)

Обучение - multitask learning
Masked Language Modeling (MLM)

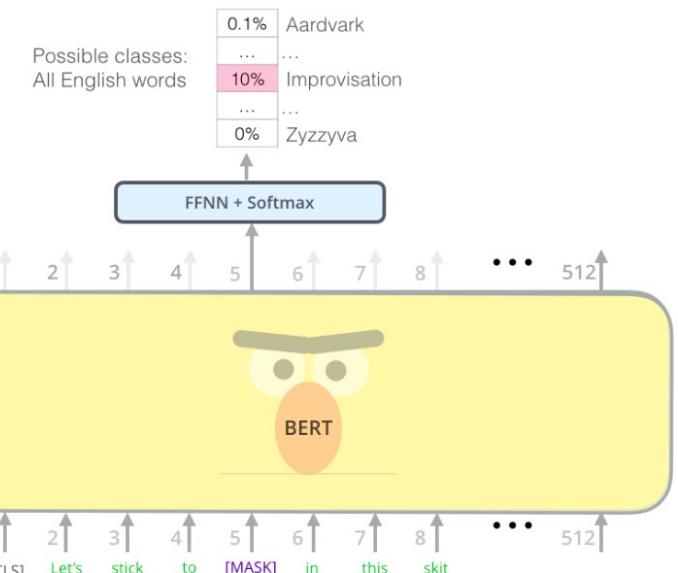
- ▶ Маскируем случайные статические токены (с заданной вероятностью 15%)
 - ▶ 80% заменяются токеном MASK
 - ▶ 10% заменяются на случайный токен
 - ▶ 10% остаются неизменными.
- ▶ Модель предсказывает верный токен, лосс считается только на этих 15% токенов.

+ Next Sentence Prediction

- Предсказываем, является ли два предложения следующими друг за другом

Masked Language Modeling (MLM)

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

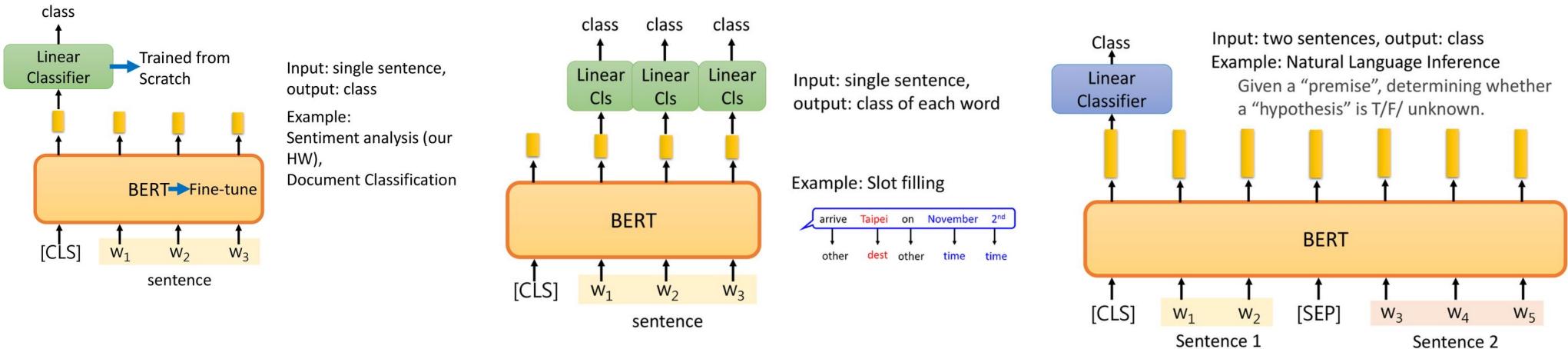


Randomly mask 15% of tokens

Input

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

BERT – примеры



- Extraction-based Question Answering (QA) (E.g. SQuAD)

Document: $D = \{d_1, d_2, \dots, d_N\}$

Query: $Q = \{q_1, q_2, \dots, q_N\}$



output: two integers (s, e)

Answer: $A = \{q_s, \dots, q_e\}$

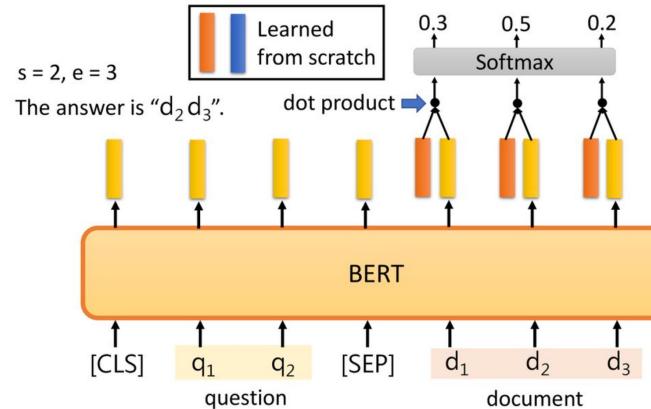
In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of rain, sleet, and snowfall are called “showers”.

What causes precipitation to fall?

gravity $s = 17, e = 17$

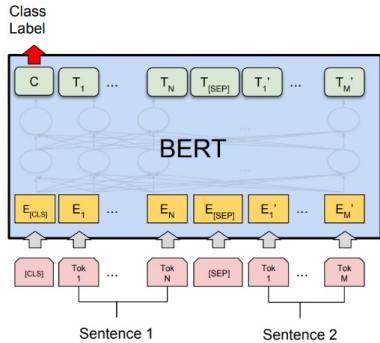
What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud $s = 77, e = 79$

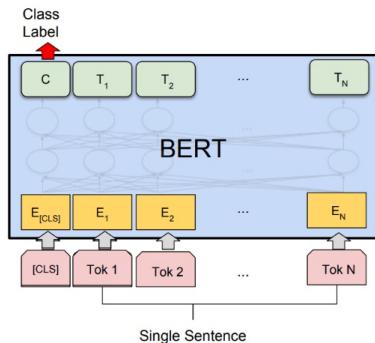


<https://slideplayer.com/slide/17025344/>

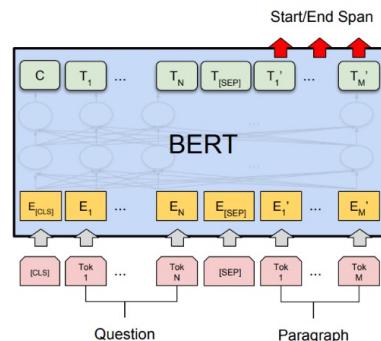
BERT – Finetuning



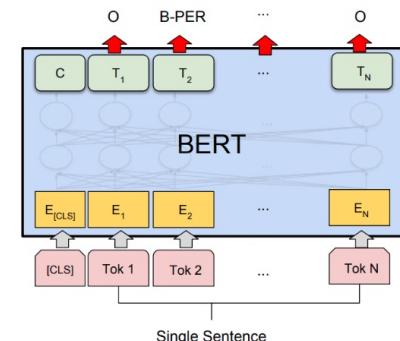
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(b) Single Sentence Classification Tasks:
SST-2, CoLA



(c) Question Answering Tasks:
SQuAD v1.1



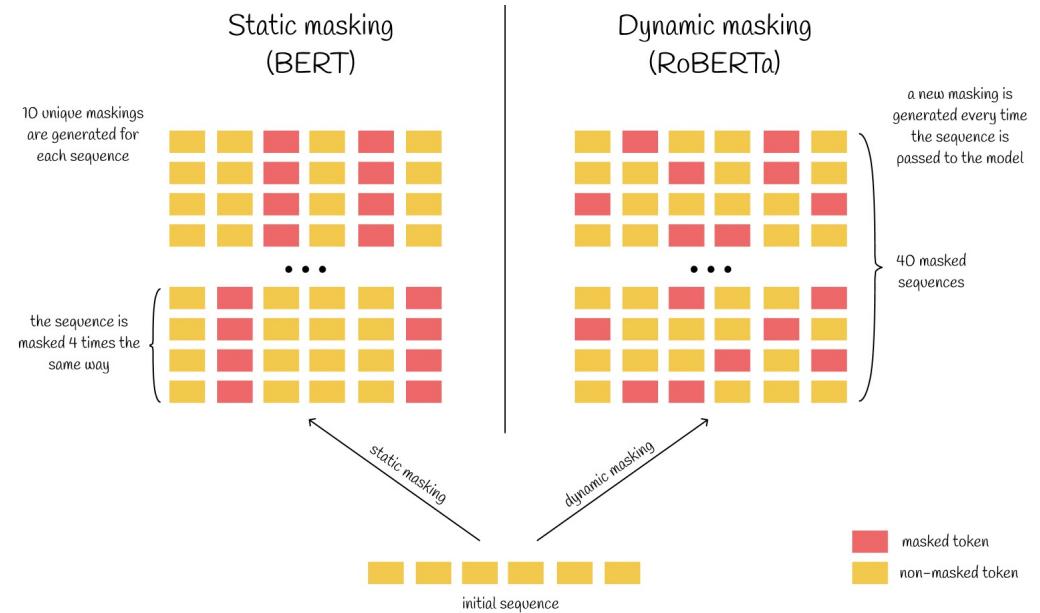
(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

- ▶ Замораживаем кодировщики.
- ▶ Обучаем только слой, который отвечает за нужный результат.
- ▶ Существенное ускорение обучения

RoBERTa

BERT был существенно недообучен
Обучение - multitask learning Masked Language Modeling (MLM) без Next Sentence Prediction

- ▶ Маскируем случайные динамические токены.
- ▶ Модель предсказывает верный токен, лосс считается только на этих 15% токенов.
- ▶ Лучший подбор гиперпараметров.



<https://towardsdatascience.com/roberta-1ef07226c8d8/>

<https://arxiv.org/abs/1907.11692>

DistilBERT

Knowledge distillation - обучение маленькой модели (ученика) воспроизводить поведение большой модели (учителя).

- ▶ DistilBERT - 6 слоев вместо 12
- ▶ Инициализация слоями обученного BERT'a
- ▶ обычный MLM лосс + лосс, измеряющий сходство выходов учителя и ученика:

$$L_{ce} = \sum_i t_i \log s_i$$

+ лосс, измеряющий похожесть скрытых состояний учителя и ученика (косинусное расстояние)

<https://www.kaggle.com/discussions/getting-started/163968>

<https://arxiv.org/abs/1910.01108v4>

<https://arxiv.org/abs/1910.01108v4>

	BERT	RoBERTa	DistilBERT	XLNet
Size (millions)	Base: 110 Large: 340	Base: 110 Large: 340	Base: 66	Base: ~110 Large: ~340
Training Time	Base: 8 x V100 x 12 days* Large: 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	Large: 1024 x V100 x 1 day; 4-5 times more than BERT.	Base: 8 x V100 x 3.5 days; 4 times less than BERT.	Large: 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	Base: 16 GB BERT data Large: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

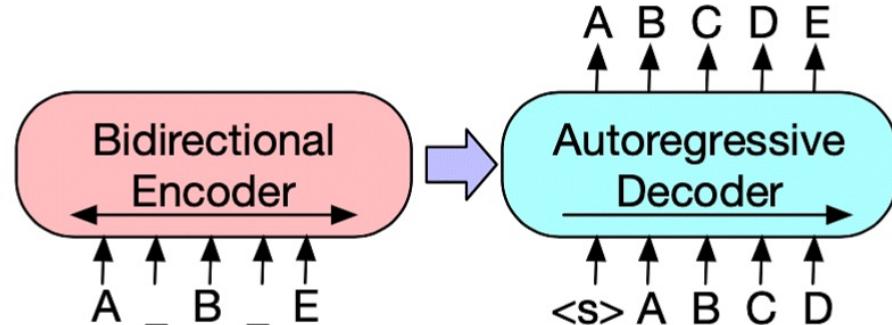


GPT-like и T5-like модели для текстов



BART (Bidirectional and Auto-Regressive Transformers)

- ▶ Transformer Encoder-Decoder
- ▶ Denoising seq2seq autoencoder, обучается на восстановлении зашумленных данных
- ▶ Text Infilling task
- ▶ Даёт возможность генерации текста с учётом понимания контекста.



Пробовали разные pretrain tasks

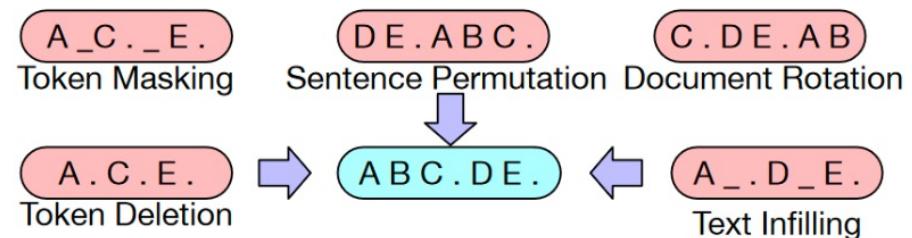
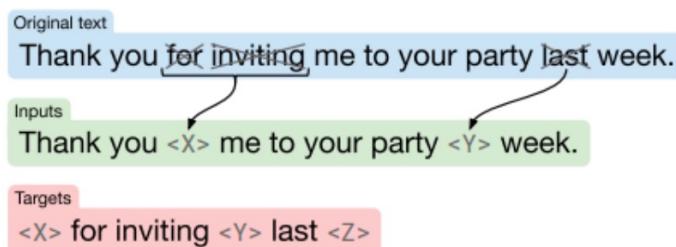
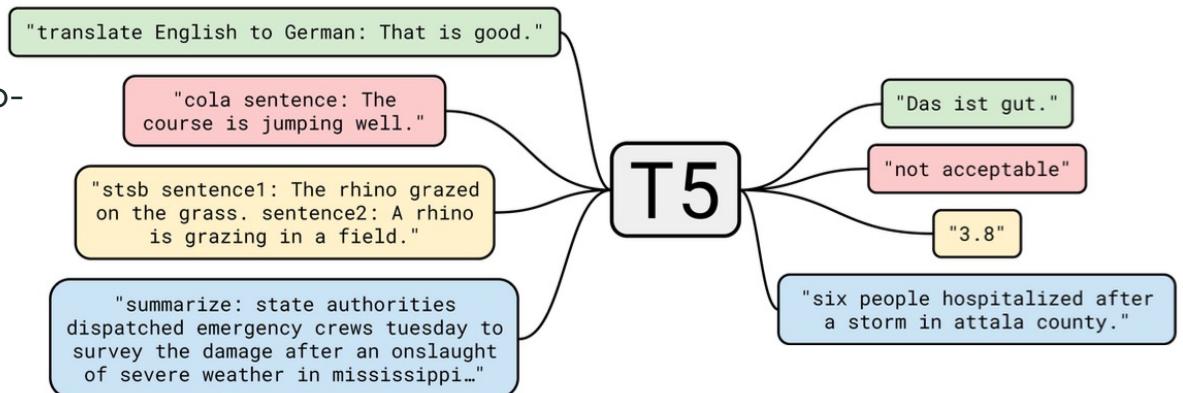


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

T5 (Text-to-Text Transfer Transformer)

- ▶ Формулируем любую задачу как text-to-text
- ▶ К входным данным добавляется текстовый префикс конкретной задачи
- ▶ Из входных данных исключается неполные предложения, обсценная лексика и тд

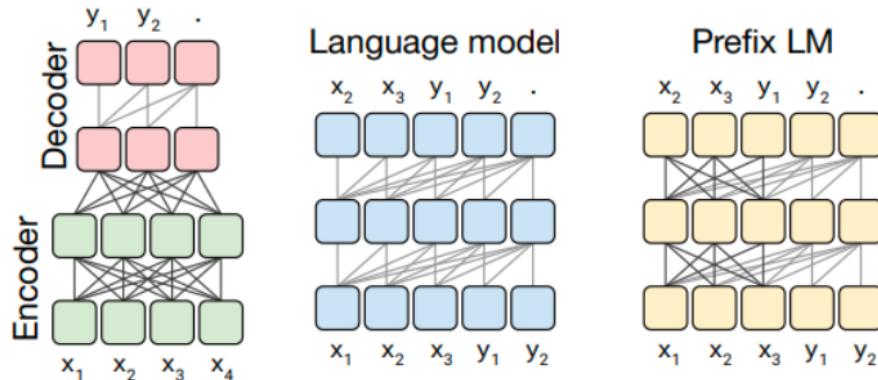


- Encoder-Decoder архитектура
- Размер до 11B параметров
- Новый большой и хороший датасет C4 (Colossal Clean Crawled Corpus)
- Supervised pretrain tasks (translation, summarization,..)
- Unsupervised pretrain task - replace span, заполнение пропусков

<https://arxiv.org/abs/1910.10683v4>

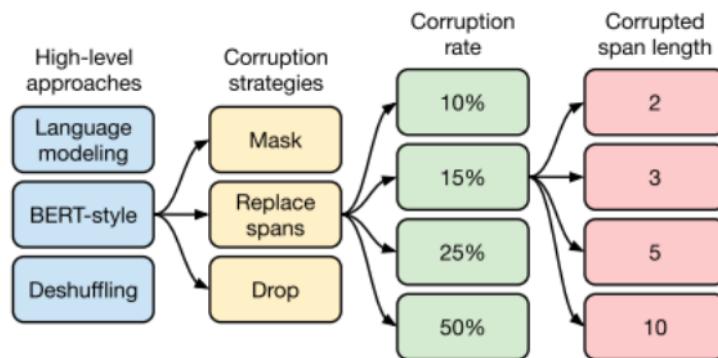
T5 (Text-to-Text Transfer Transformer)

Попробовали разные архитектуры



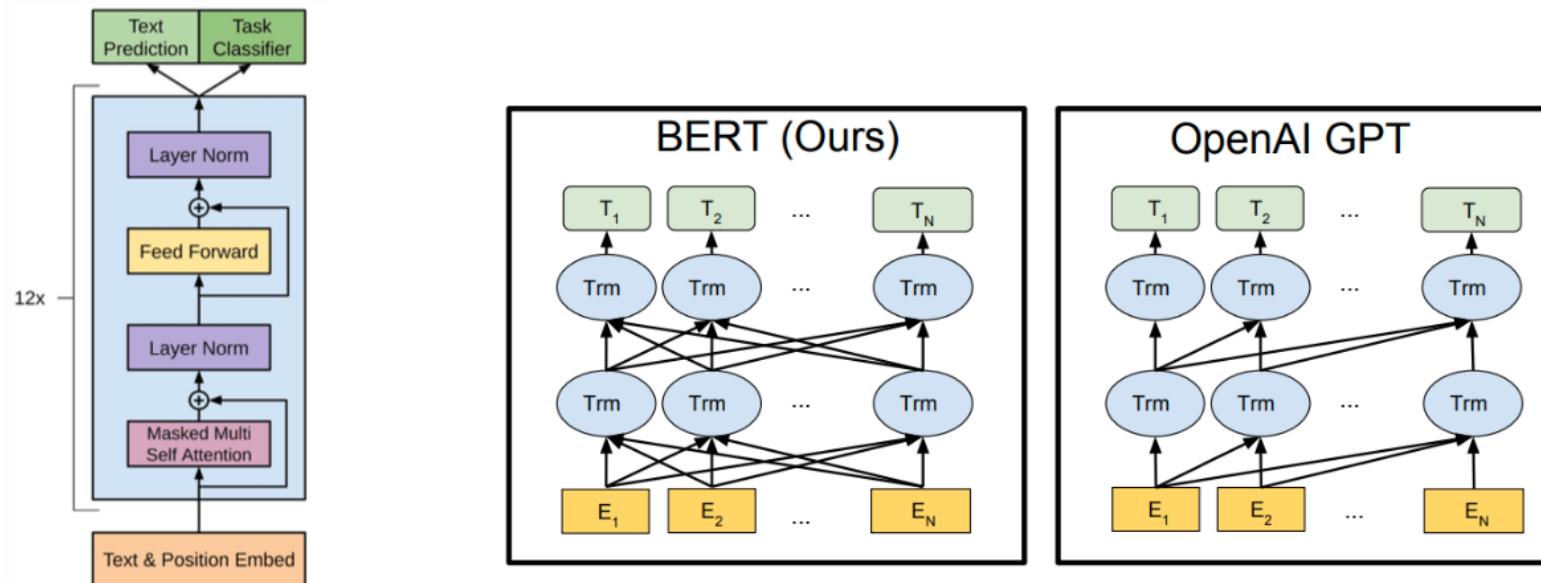
▶ Итогом стала классическая кодировщик-декодировщик архитектура с 11B параметров.

Сравнили разных варианты unsupervised pretrain tasks



Generative Pre-trained Transformer (GPT)

- Transformer Decoder
- Просто предсказываем следующий токен



GPT Family

Последовательное увеличение количества данных, размера моделей и вычислений

GPT-1

- Radford A. et al. (2018) Improving language understanding by generative pre-training.
- 117M параметров, L=12, H=768, A=12
- BooksCorpus dataset, 4.5 ГБ текста, 7K книг
- Размер контекста 512 токенов

GPT-2

- Radford A. et al. (2019) Language models are unsupervised multitask learners.
- 1.5B параметров, L=48, H=1600
- WebText dataset, 40 ГБ текста, 8М веб-страниц
- Размер контекста 1024 токенов

GPT Family

GPT-3

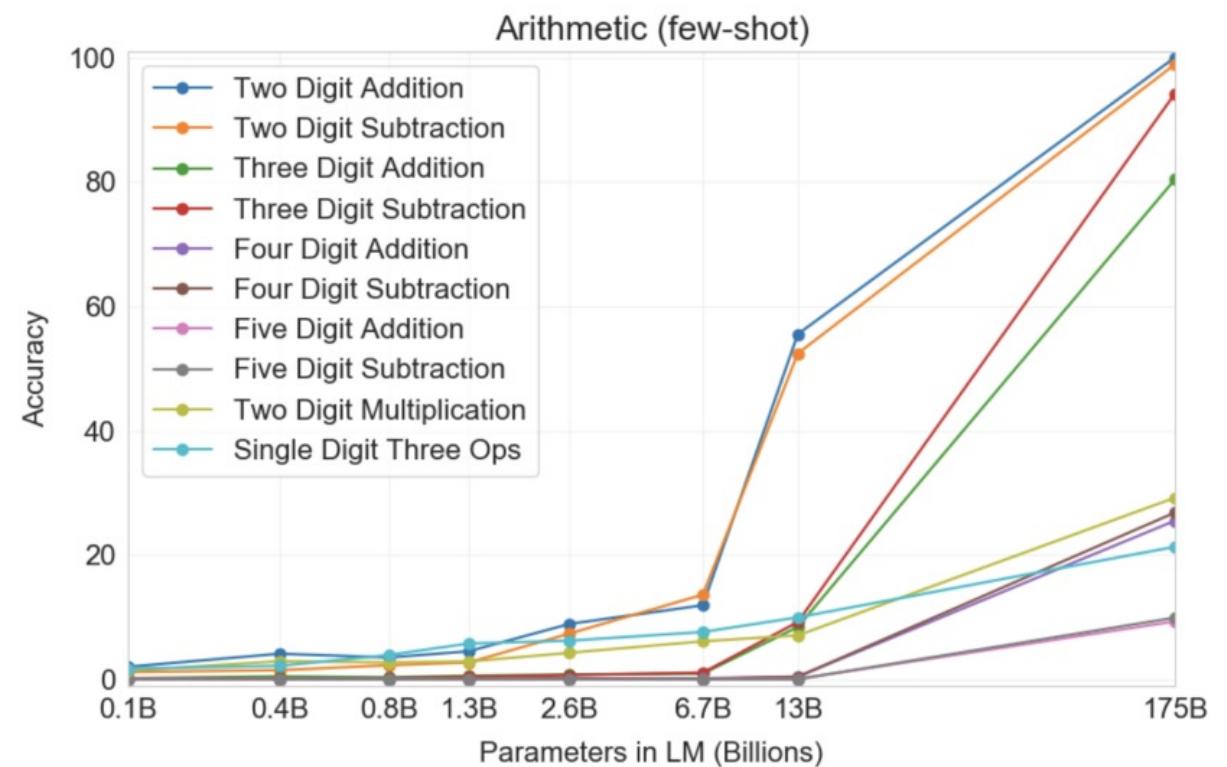
- Brown T. et al. (2020) Language models are few-shot learners.
- 175B параметров, L=96, H=12288, A=96
- 570ГБ текста, 300B токенов (Common Crawl, WebText, ..)
- Размер контекста 2048 токенов

GPT-4

- ???
- Размер контекста 8192 и 32768 токенов
- Мультимодальная модель (работает также с изображениями)

GPT Family

- ▶ На текущий момент, большее количество параметров даёт лучшее решение.



Zero-shot learning

- ▶ Использование модели совсем без дообучения!
- ▶ Язык - универсальный интерфейс
- ▶ Форматируем вход (prompt) так, чтобы была понятна задача
- ▶ Prompt engineering - новая парадигма

Zero-shot

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

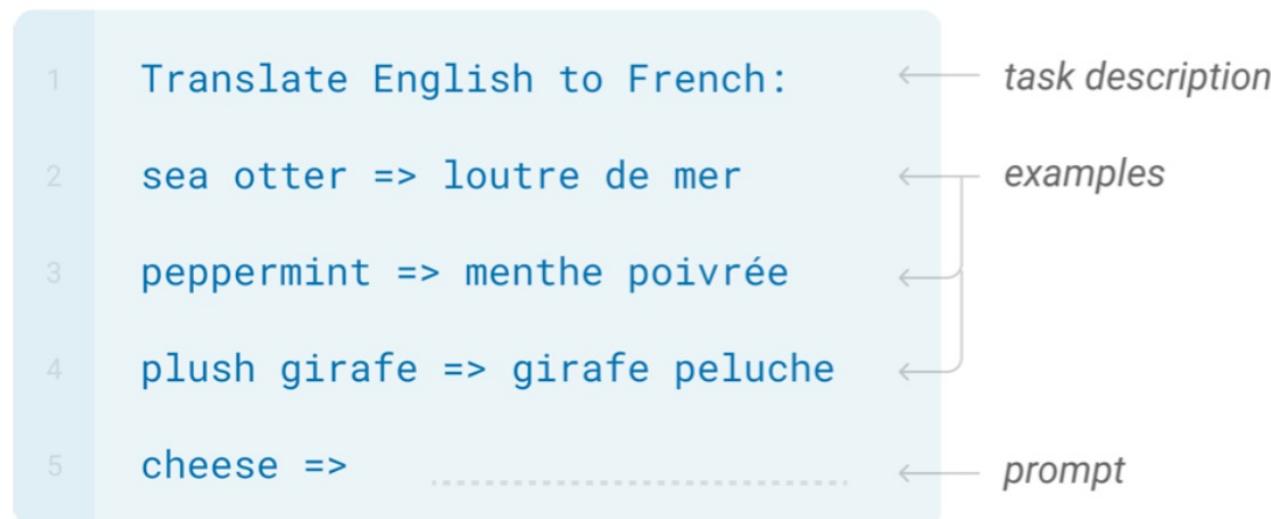


Few-shot learning

Без дообучения, но показываем модели несколько примеров
того, как решать задачу

Few-shot

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



Few-shot learning

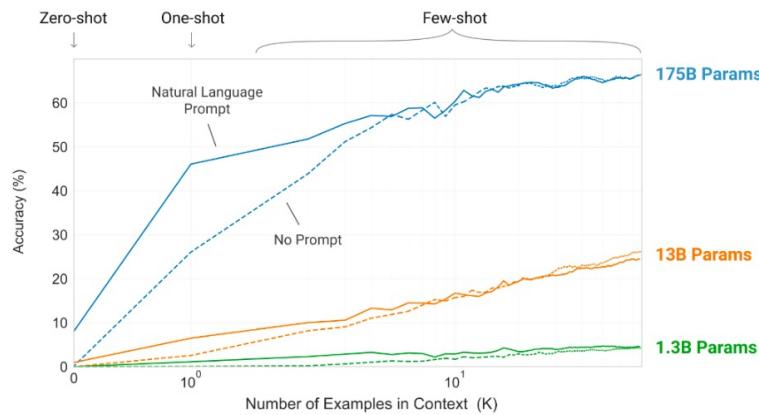


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper “in-context learning curves” for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation.

Ричард Саттон

<http://www.incompleteideas.net/Incldeas/BitterLesson.html>

Законы масштабирования

N - количество параметров модели
D - объём обучающих данных
C - вычислительный бюджет

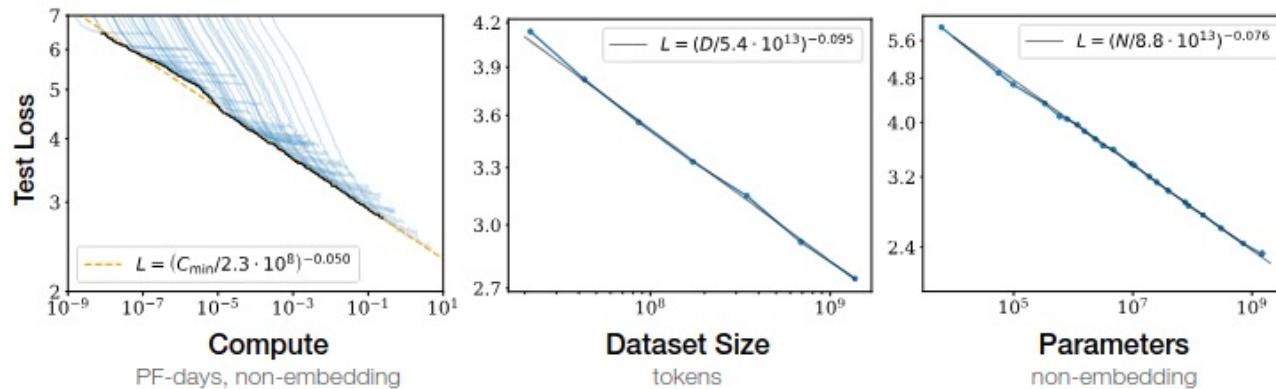


Figure 1 Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute^[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

Два режима:

Variance-limited — доминирует при малых данных, где увеличение D снижает дисперсию оценки.

Resolution-limited — актуален для больших моделей, где производительность ограничена способностью сети "разрешать" тонкие паттерны в данных.

<https://arxiv.org/abs/2001.08361>

<https://research.google/pubs/explaining-neural-scaling-laws/>

Chinchilla Scaling Laws - Закон оптимальности Шиншиллы

Efficient frontier. We can approximate the functions N_{opt} and D_{opt} by minimizing the parametric loss \hat{L} under the constraint $\text{FLOPs}(N, D) \approx 6ND$ (Kaplan et al., 2020). The resulting N_{opt} and D_{opt} balance the two terms in Equation (3) that depend on model size and data. By construction, they have a power-law form:

$$N_{opt}(C) = G \left(\frac{C}{6} \right)^a, \quad D_{opt}(C) = G^{-1} \left(\frac{C}{6} \right)^b, \quad \text{where} \quad G = \left(\frac{\alpha A}{\beta B} \right)^{\frac{1}{\alpha+\beta}}, \quad a = \frac{\beta}{\alpha + \beta}, \quad \text{and} \quad b = \frac{\alpha}{\alpha + \beta}. \quad (4)$$

We show contours of the fitted function \hat{L} in Figure 4 (left), and the closed-form efficient computational frontier in blue. From this approach, we find that $a = 0.46$ and $b = 0.54$ —as summarized in Table 2.

Последствия:

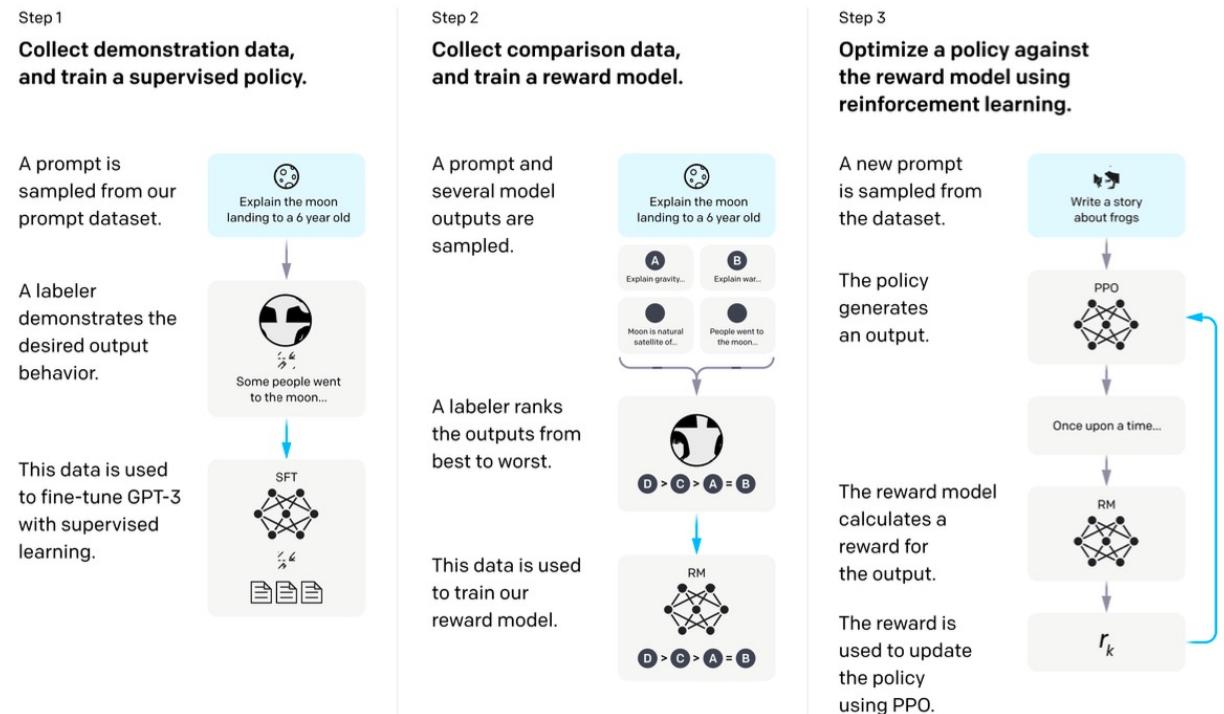
- ▶ Переобучение существующих моделей:
- ▶ Экономия ресурсов
- ▶ Претрейн обучение на одну эпоху

<https://arxiv.org/abs/2203.15556>

<https://llmstudio.ru/blog/pretrain-llm-scaling-laws>

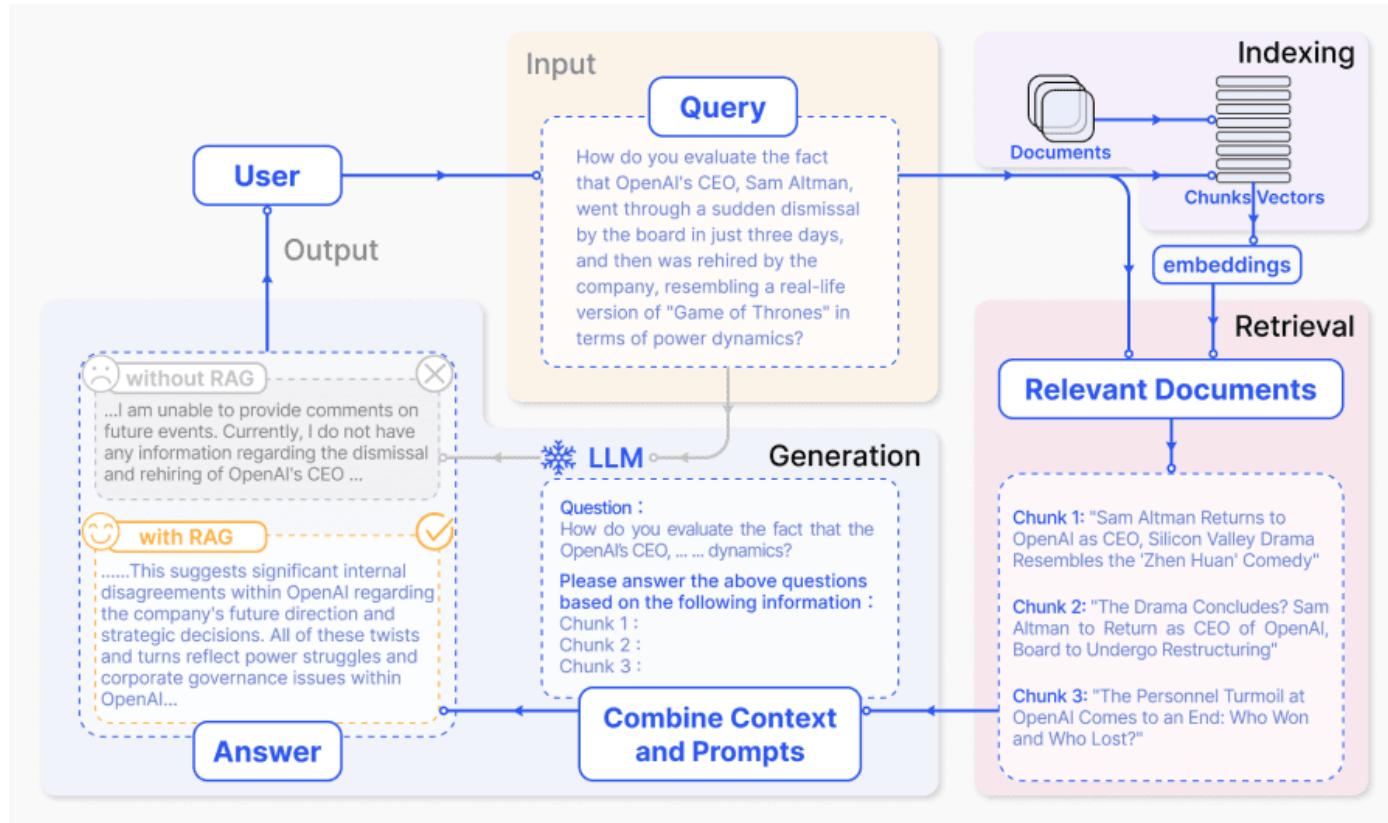
ChatGPT

- ▶ Supervised finetuning на примерах правильного поведения
- ▶ RLHF (Reinforcement Learning from Human Feedback)



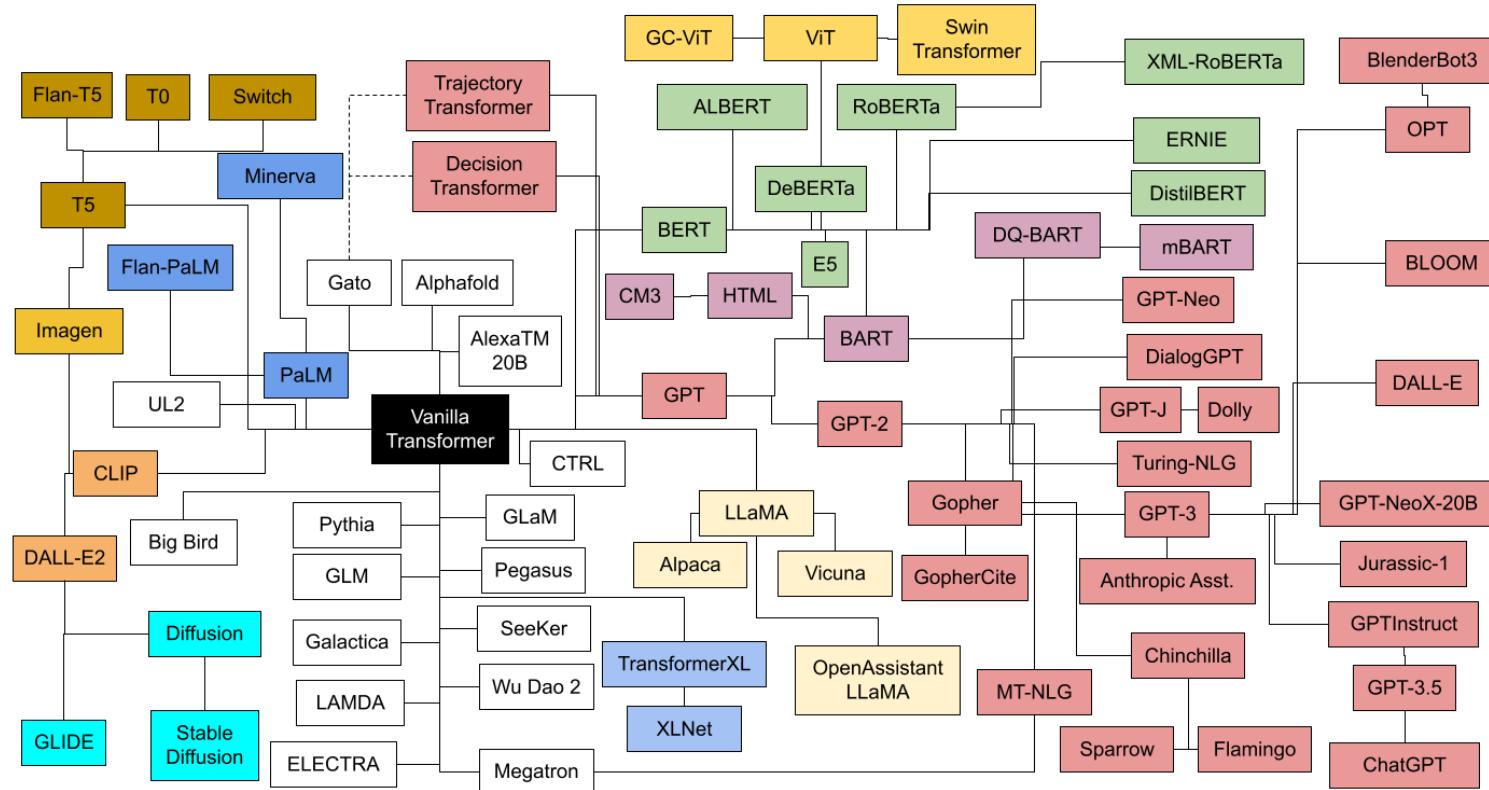
<https://openai.com/index/instruction-following/>
<https://arxiv.org/abs/2203.02155>

Retrieval Augmented Generation



<https://www.promptingguide.ai/research/rag>

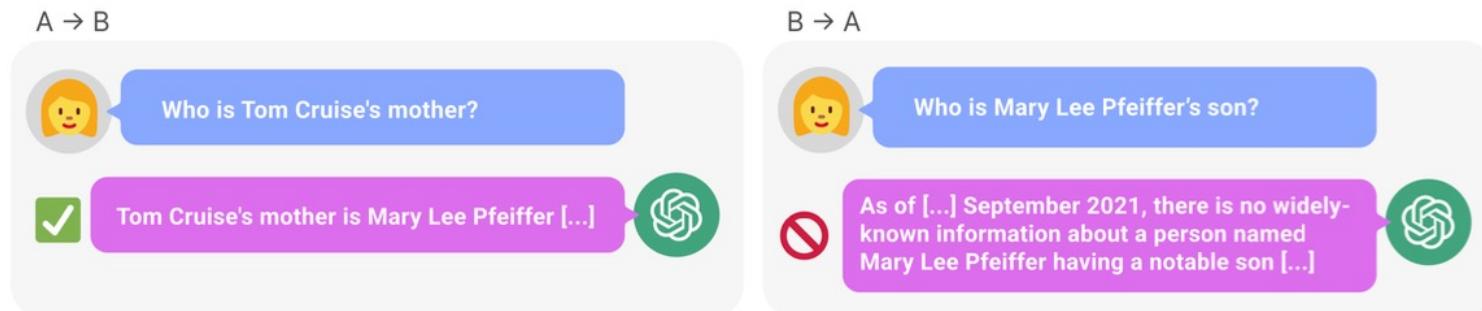
Зоопарк трансформеров



<https://amatria.in/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/>

Large Language Models

Large Language Models - stochastic parrots or AGI?



Berglund L. et al. The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A"

<https://dl.acm.org/doi/10.1145/3442188.3445922>

Заключение

- ▶ 2018 год – начало взрывного развития языковых моделей
- ▶ На текущий момент определяет количество параметров и размер выборки
- ▶ Опасность – недостаток данных и генерированные данные в тренировке