

Seq2seq & Attention

2024

Seq2seq

- Seq2seq (Sequence to Sequence) – задача преобразования одной последовательности в другую последовательность
- Применение:
 - машинный перевод (machine translation)
 - чат-боты (chat bots)
 - вопросно-ответные системы (question-answering systems)
 - автоматическое реферирование (summarization)
 - генерация подписей к изображениям (image captioning)
 - ...

Seq2seq: исходные статьи

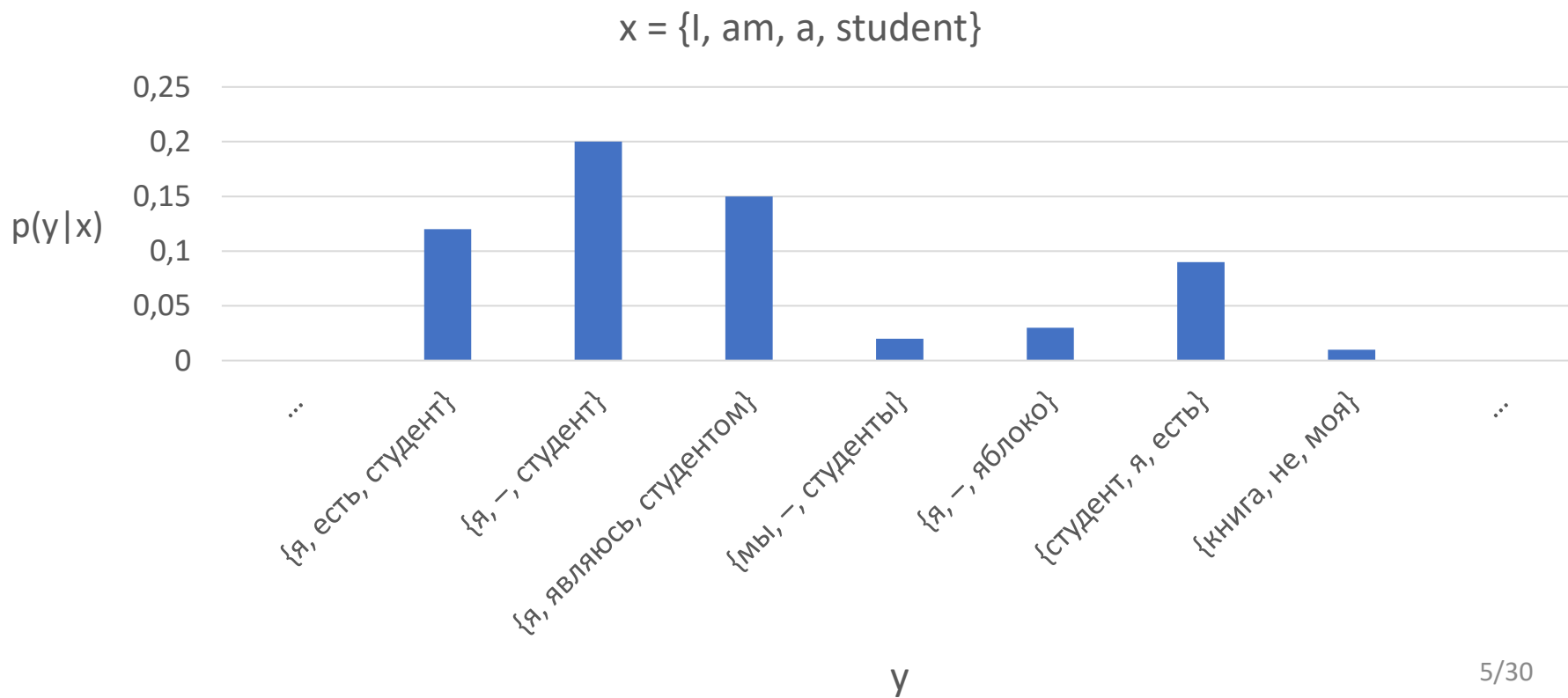
- Kalchbrenner Nal, Blunsom Phil – Recurrent continuous translation models
// Conference on Empirical Methods in Natural Language Processing (EMNLP). 2013
 - Recurrent Continuous Translation Model: отображение входной последовательности в вектор с помощью n-gram CNN, а затем – в выходную последовательность на основе комбинации n-gram CNN и RNN
- Cho Kyunghyun, van Merriënboer Bart, Gulcehre Caglar, Bahdanau Dzmitry, Bougares Fethi, Schwenk Holger, Bengio Yoshua – Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation // Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014
 - RNN Encoder-Decoder как дополнение Phrase-based SMT
- Sutskever Ilya, Vinyals Oriol, Le Quoc V. – Sequence to Sequence Learning with Neural Networks // NeurIPS-2014
 - 4-layer LSTM в энкодере и декодере для машинного перевода без использования Phrase-based SMT

Seq2seq

- Постановка задачи:
 - input sequence: $x = \{x_1, \dots, x_m\}$
 - output sequence: $y = \{y_1, \dots, y_n\}$
 - task: $y^* = \arg \max_y p(y|x)$
 - task with model parameters: $y^* = \arg \max_y p(y|x, \theta)$

Seq2seq

- Основная функция языковой модели – моделирование вероятностного распределения последовательностей



Seq2seq

- Language model:

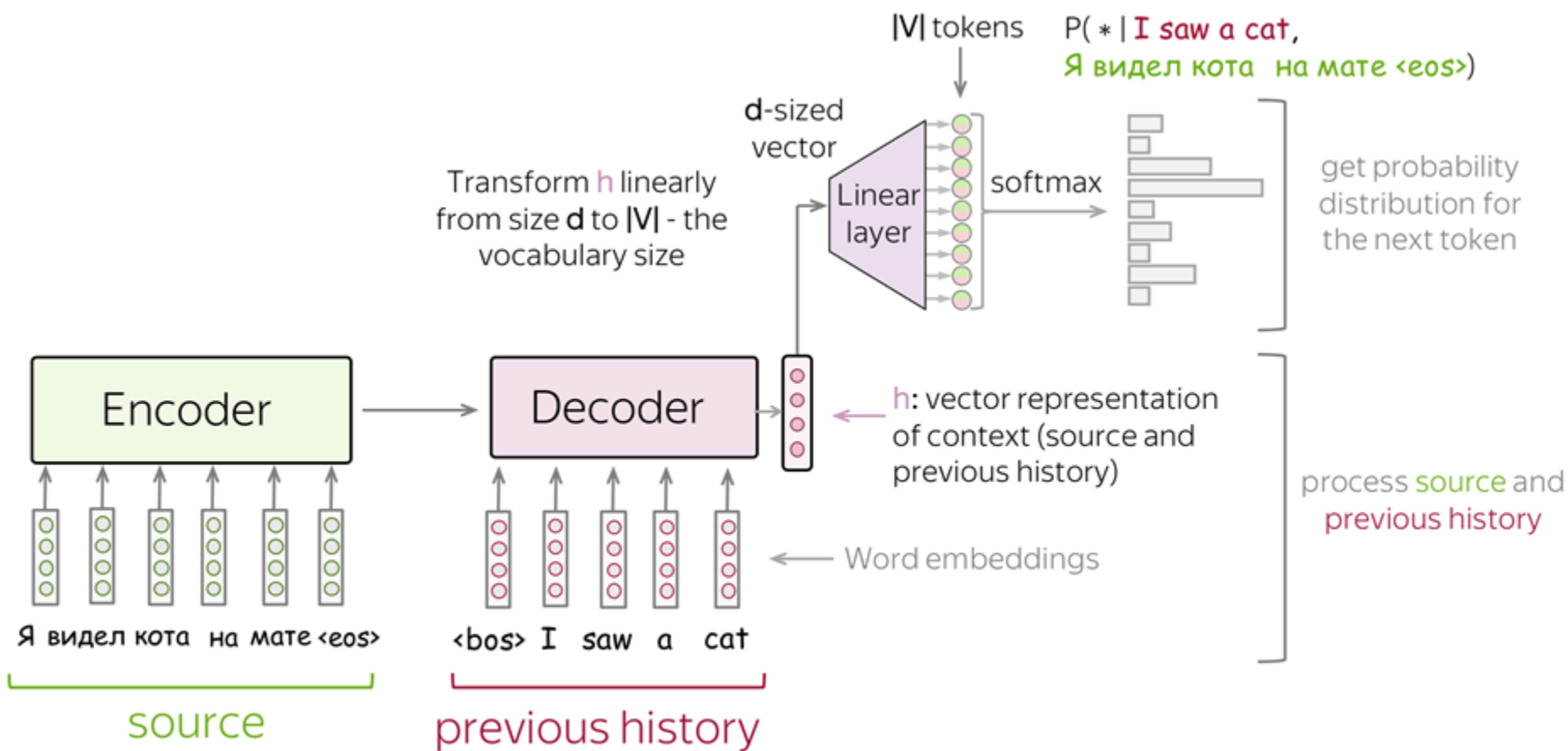
$$P(y_1, \dots, y_n) = \prod_{t=1}^n p(y_t | y_{<t})$$

- Conditional language model:

$$P(y_1, \dots, y_n | \mathbf{x}) = \prod_{t=1}^n p(y_t | y_{<t}, \mathbf{x})$$

- Основной вид моделей: encoder-decoder

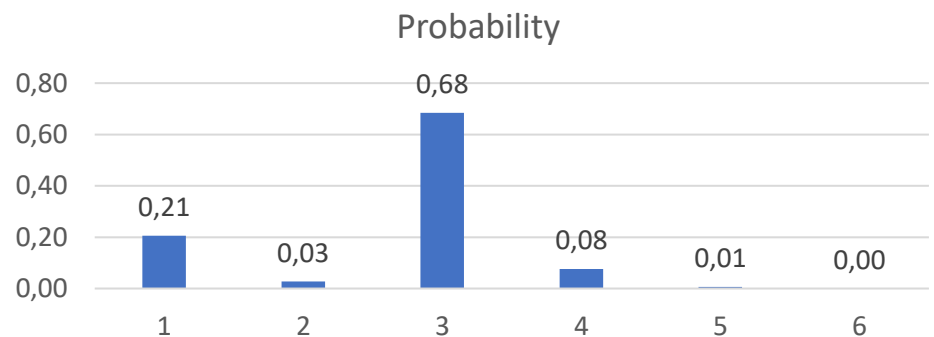
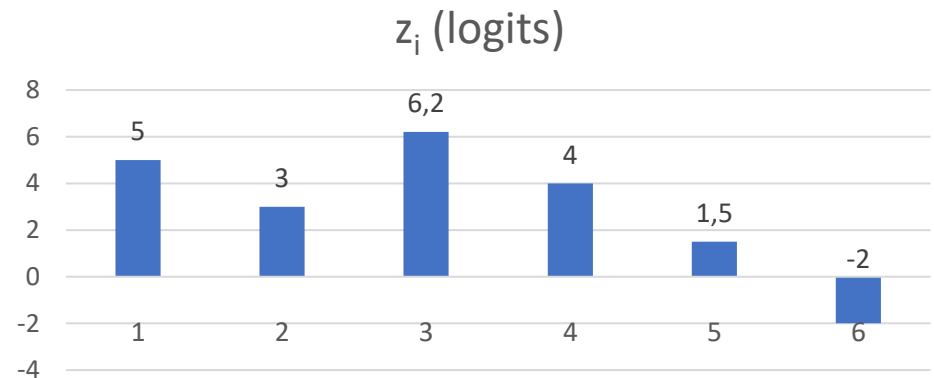
Seq2seq



SoftMax

$$\text{SoftMax}(z_1, \dots, z_n) = \left(\frac{e^{z_1}}{\sum_{k=1}^n e^{z_k}}, \dots, \frac{e^{z_n}}{\sum_{k=1}^n e^{z_k}} \right)$$

N	z_i (logits)	Exp	Probability
1	5	148.4	0.21
2	3	20.1	0.03
3	6.2	492.7	0.68
4	4	54.6	0.08
5	1.5	4.5	0.01
6	-2	0.1	0.00
		720.5	1.00

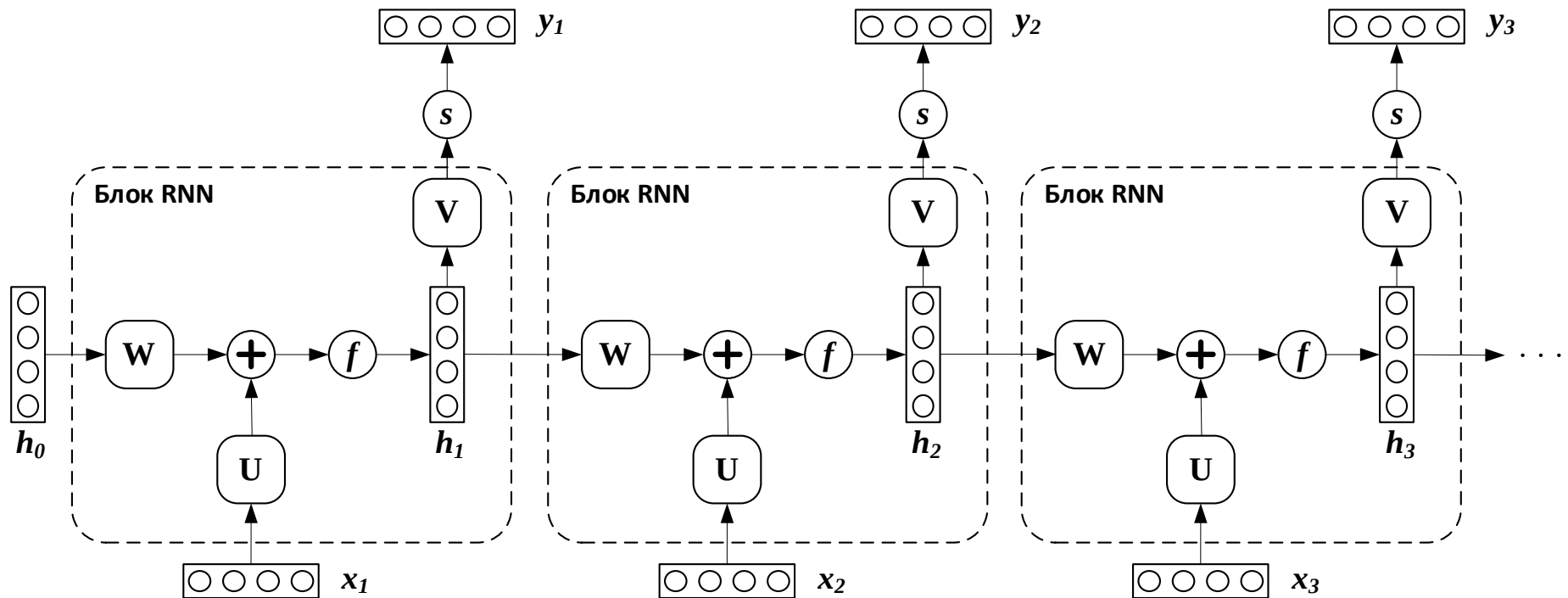


Рекуррентные сети: Simple (Vanilla) RNN

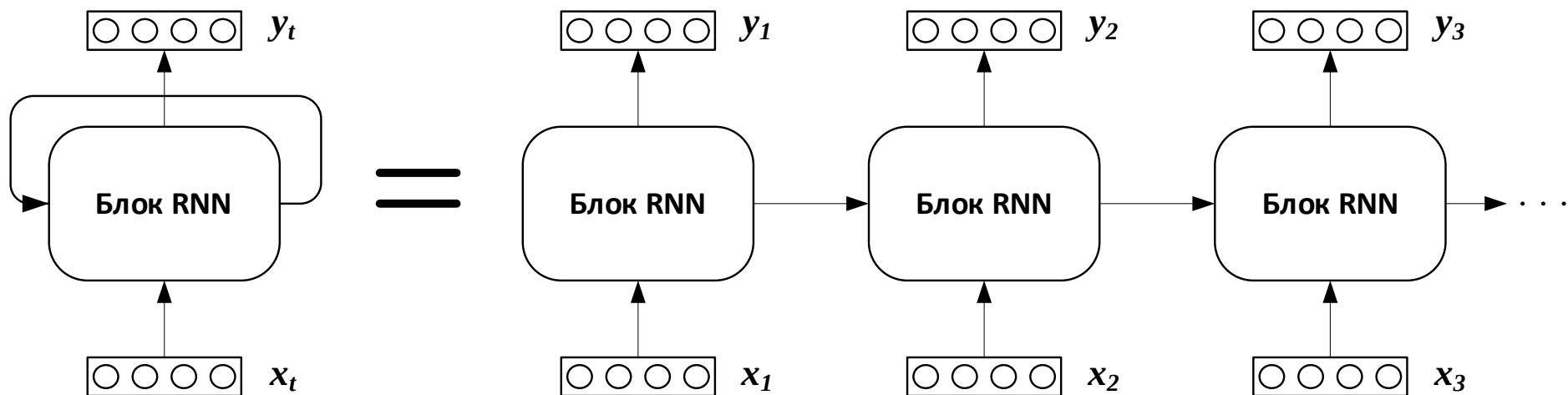
$$\vec{h}_t = f(\mathbf{W}\vec{h}_{t-1} + \mathbf{U}\vec{x}_t + \vec{b}) \quad \vec{y}_t = s(\mathbf{V}\vec{h}_t + \vec{c})$$

f – σ , tanh, ReLU

s – softmax

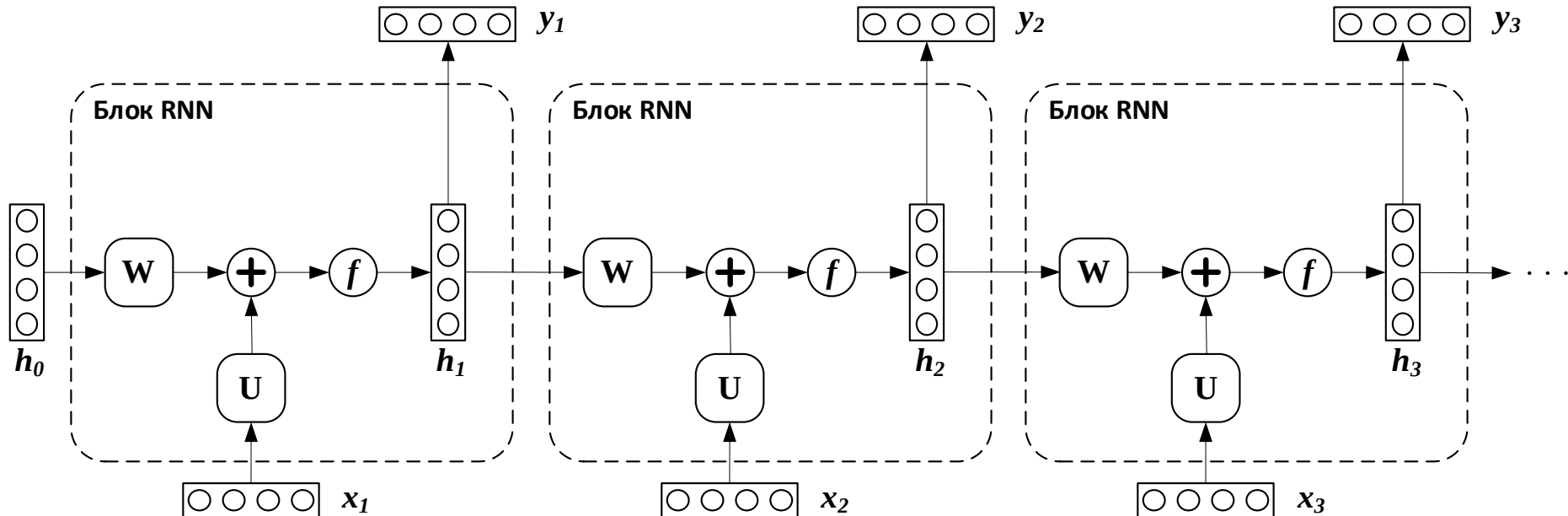


Рекуррентные сети: Simple RNN

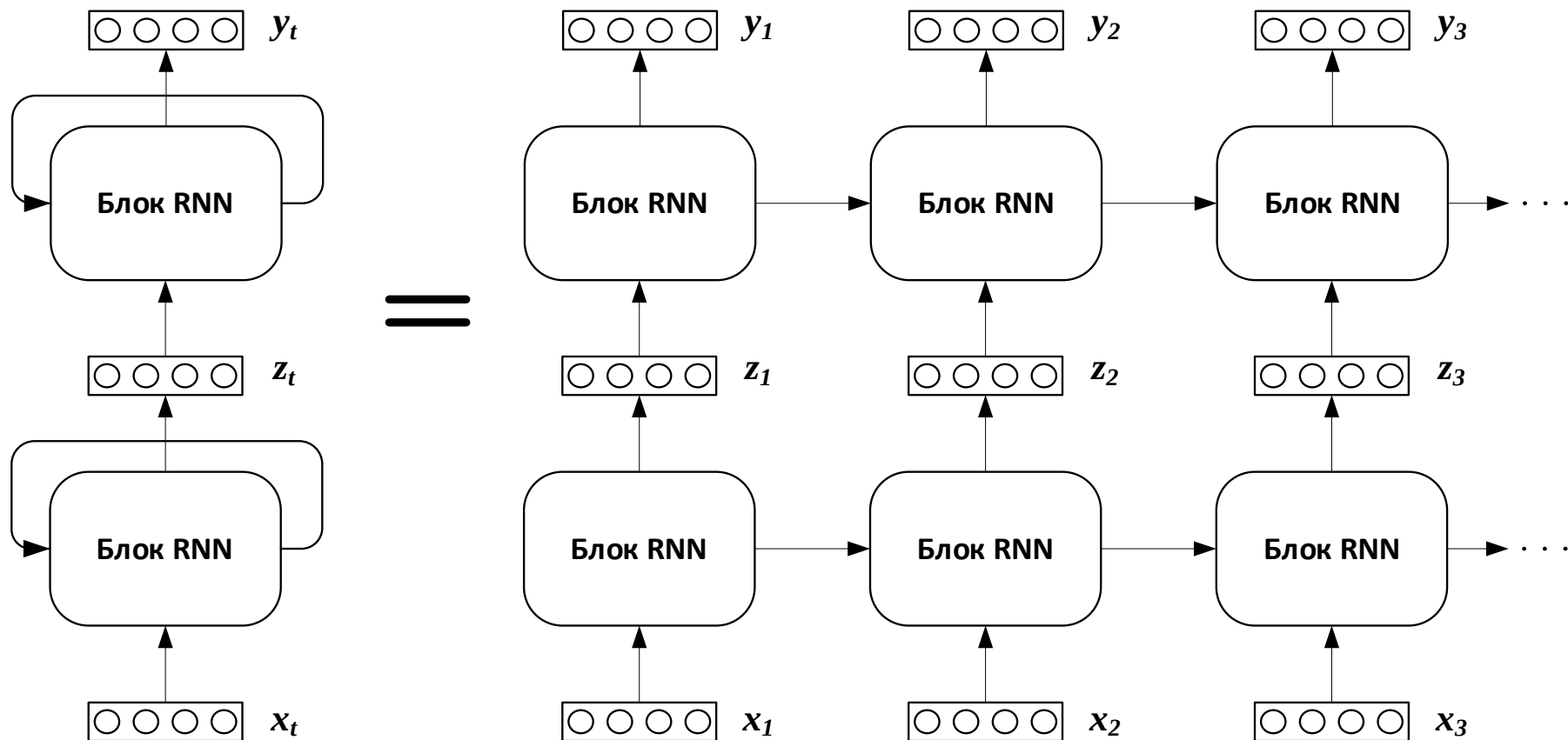


Рекуррентные сети: Simple RNN

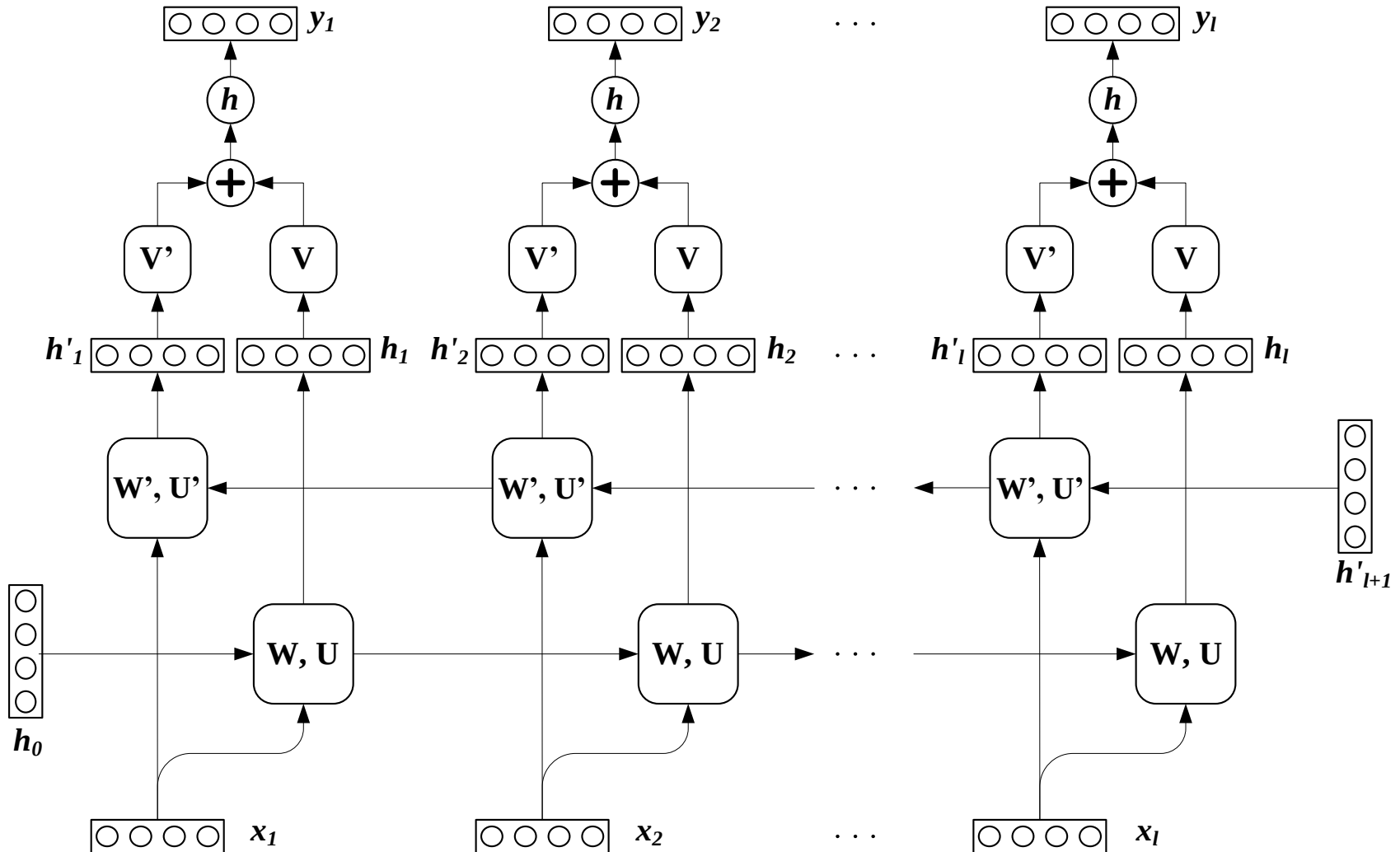
$$\vec{h}_t = f(W\vec{h}_{t-1} + U\vec{x}_t + \vec{b}) \quad \vec{y}_t = \vec{h}_t \quad f - \tanh, \text{ReLU}$$



Рекуррентные сети: двухслойная RNN



Двунаправленные рекуррентные сети (Bidirectional RNN)



Seq2seq: вектор скрытого состояния кодировщика

- Вектор скрытого состояния для RNN на шаге t :

$$h_t = f(x_t, h_{t-1})$$

- Вектор скрытого состояния на выходе кодировщика (в общем случае):

$$c = g(\{h_1, \dots, h_{T_x}\})$$

- Вектор скрытого состояния на выходе кодировщика (часто):

$$c = h_{T_x}$$

Seq2seq: декодер

- Условная вероятность слова y_t (см. следующий слайд):

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = \text{SoftMax}(q(y_{t-1}, h_t, c))$$

где q – нелинейная, потенциально многослойная функция:
выходная размерность совпадает с размерностью словаря,

h_t – вектор скрытого состояния декодера на шаге t ,

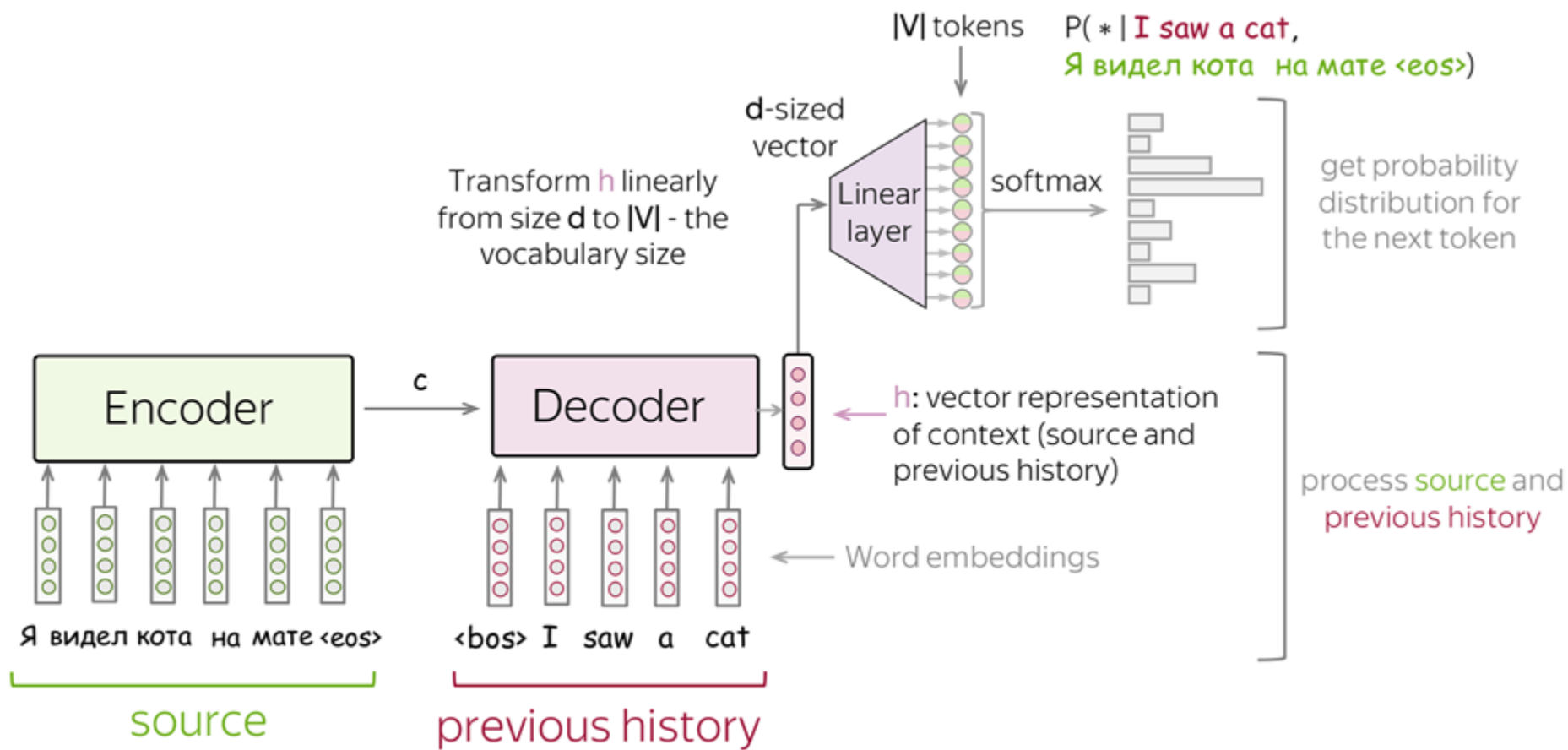
c – вектор скрытого состояния на выходе кодировщика

- Как правило, вектор c используется только для инициализации скрытого состояния декодера:

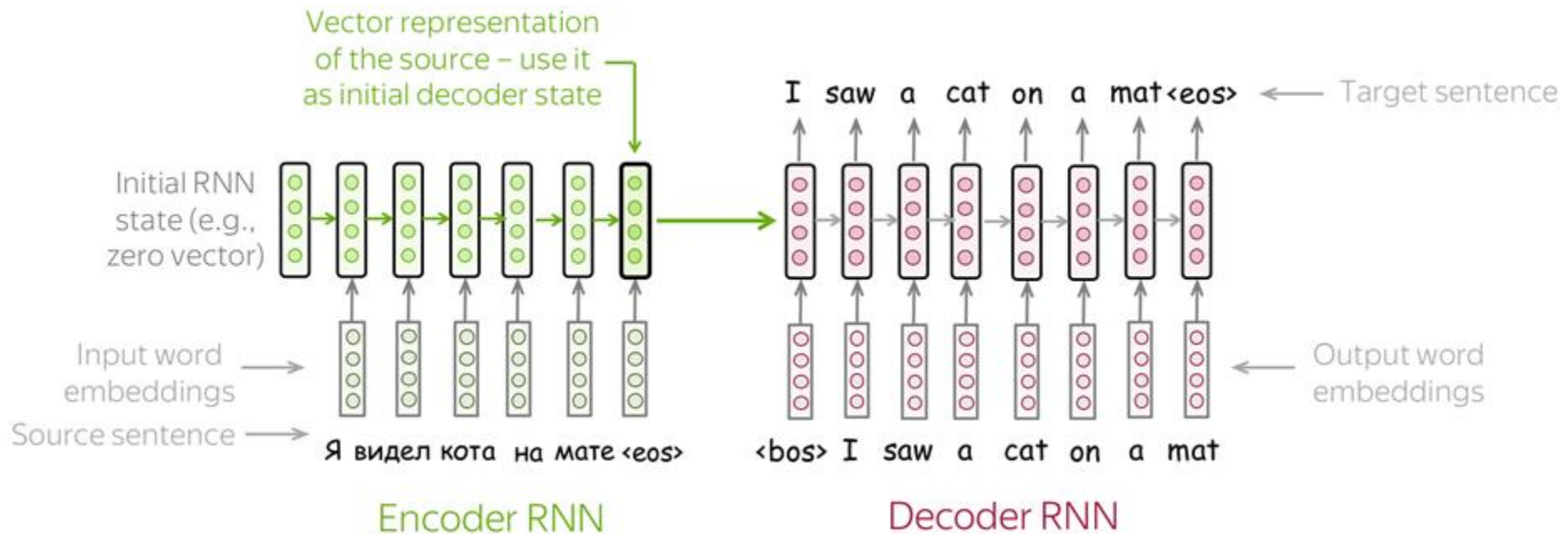
$$h_t = r(y_{t-1}, h_{t-1}), h_0 = c, y_0 = \langle eos \rangle | \langle bos \rangle$$

Seq2seq

$$h_t = r(y_{t-1}, h_{t-1}), h_0 = c$$

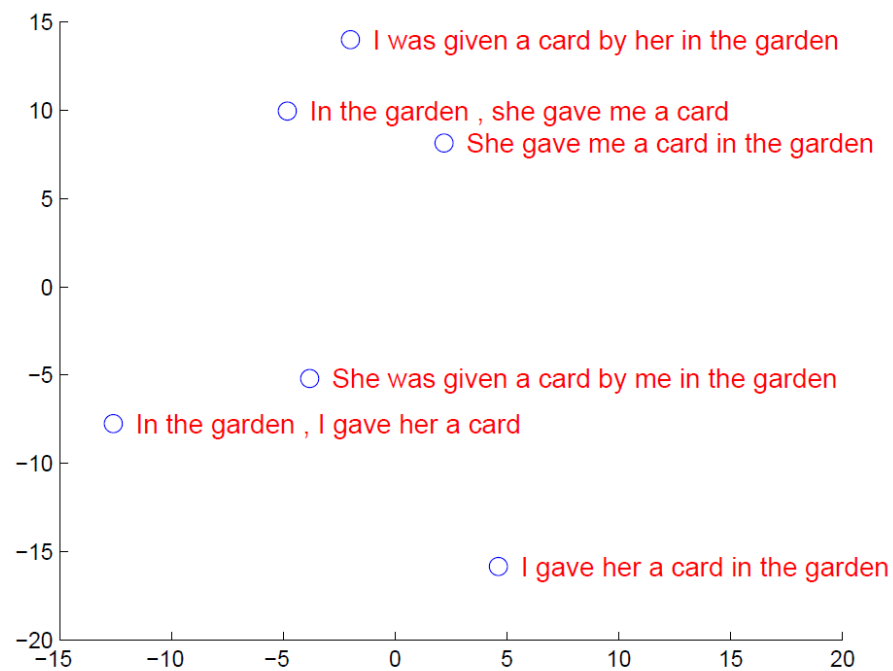
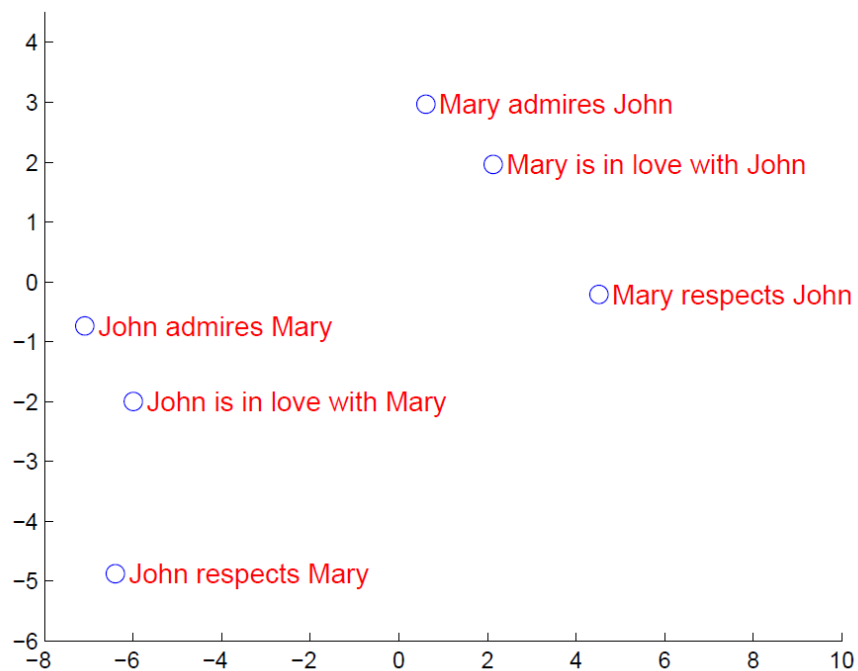


Seq2seq: RNN (LSTM)



Seq2seq

- Визуализация последнего состояния энкодера [Sutskever et al., 2014]:



Seq2seq: обучение

- Обучающий пример: $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_n)$
- Шаг t : модель предсказывает вероятности для всех слов словаря:

$$p^{(t)} = p(* | y_1, \dots, y_{t-1}, x_1, \dots, x_m)$$

- Значение целевой переменной: $p^* = \text{onehot}(y_t)$
 - 1 – правильный токен y_t
 - 0 – остальные токены

- Функция потерь (cross-entropy):

$$\text{Loss}(p^*, p) = - \sum_{i=1}^V p_i^* \log(p_i) = -p^* \log(p) = -\log(p_{y_t}) = -\log(p(y_t | y_{<t}, x))$$

- Для всего примера:

$$- \sum_{t=1}^n \log(p(y_t | y_{<t}, x))$$

Seq2seq: обучение

Source sequence:

Я видел кота на мате <eos>

Target sequence:

I saw a cat on a mat <eos>

← one training example

← one step for this example

previous tokens

we want the model
to predict this

Model prediction: $p(* | \text{I saw a, Я ... <eos>})$

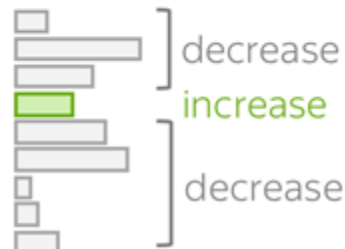


← cat →

Target



Loss = $-\log(p(\text{cat})) \rightarrow \min$



Attention

- Весь вход сжимается энкодером в вектор фиксированной размерности
 - для энкодера сложно сжать всю информацию в такой вектор – проблема при обработке длинных последовательностей
 - для декодера на разных шагах разная информация может быть релевантной
- Идея «внимания»: позволить модели на разных шагах фокусироваться на разных частях входа

Attention: исходные статьи

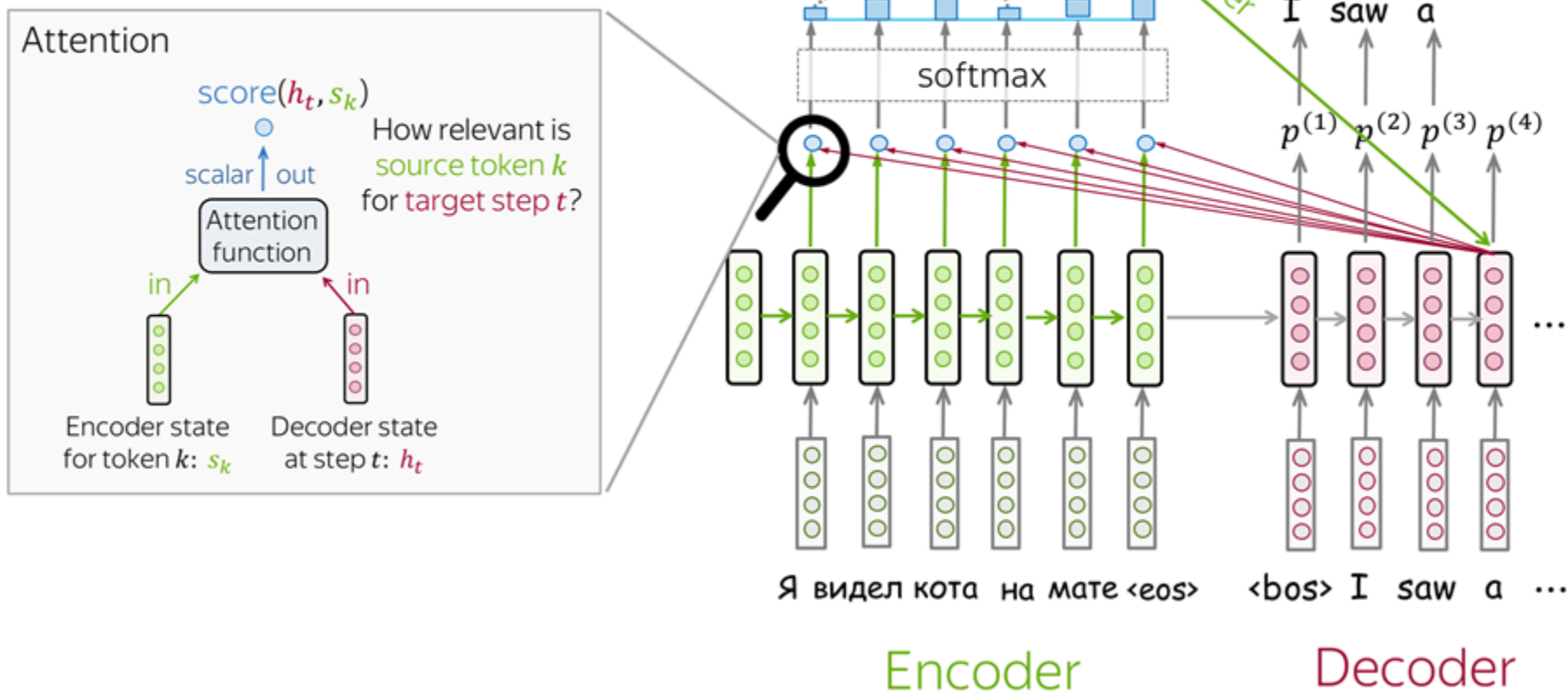
- Alex Graves – Generating Sequences With Recurrent Neural Networks // arXiv:1308.0850. 2013
- Bahdanau Dzmitry, Cho Dzmitry, Bengio Yoshua – Neural Machine Translation by Jointly Learning to Align and Translate // 3rd International Conference on Learning Representations (ICLR). 2015
- Luong Minh-Thang, Pham Hieu, Manning Christopher D. – Effective Approaches to Attention-based Neural Machine Translation // Conference on Empirical Methods in Natural Language Processing (EMNLP). 2015
- Но: идея внимания появилась в статьях 2010-2011 гг. по обработке изображений
 - Larochelle H., Hinton G.E. Learning to Combine Foveal Glimpses with a Third-Order Boltzmann Machine // Advances in Neural Information Processing Systems 23 / Curran Associates, Inc., 2010. — P. 1243–1251
 - Denil M., Bazzani L., Larochelle H., de Freitas N. Learning where to Attend with Deep Architectures for Image Tracking // arXiv, 2011. <http://arxiv.org/abs/1109.3737>

Attention

Attention output: weighted sum of encoder states with attention weights

Attention weights: distribution over source tokens

A model can learn to “pay attention” to the most relevant source tokens for each step



Attention: виды

- По размеру контекста [Luong et al., 2015]:
 - *глобальное* – внимание вычисляется по отношению ко всем входным словам
 - *локальное* – внимание вычисляется только для подмножества входных слов
- По количеству векторов входных слов, участвующих в итоговой функции внимания [Xu et al., 2015]:
 - *мягкое* (soft) – участвуют все вектора с учетом весов
 - *жесткое* (hard) – выбирается только один вектор

Xu K., Ba J., Kiros R., Cho K., Courville A.C., Salakhutdinov R., Zemel R.S., Bengio Y.
Show, attend and tell: Neural image caption generation with visual attention // ICML-2015

Attention

- Функции внимания:
 - скалярное произведение (dot-product)
 - билинейная (bilinear) – Luong attention
 - многослойный персептрон (multi-layer perceptron) – Bahdanau attention

Dot-product

$$h_t^T \times s_k$$

$$\text{score}(h_t, s_k) = h_t^T s_k$$

Bilinear

$$h_t^T \times [W] \times s_k$$

$$\text{score}(h_t, s_k) = h_t^T W s_k$$

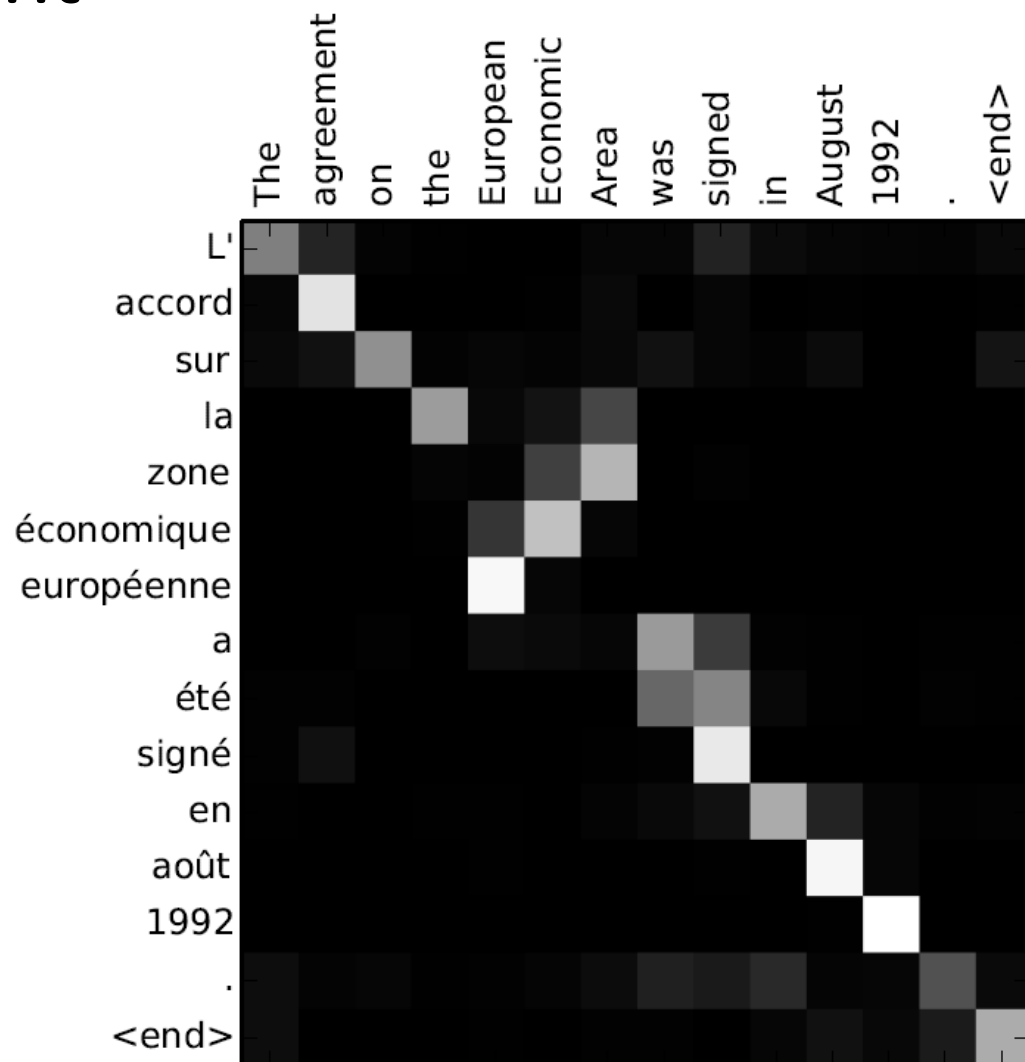
Multi-Layer Perceptron

$$w_2^T \times \tanh \left[[W_1] \times \begin{bmatrix} h_t \\ s_k \end{bmatrix} \right]$$

$$\text{score}(h_t, s_k) = w_2^T \cdot \tanh(W_1 [h_t, s_k])$$

Attention: alignment

Внимание распределяется
правильно при переводе
«zone économique
européenne» →
«European Economic Area»
[Bahdanau et al., 2014]

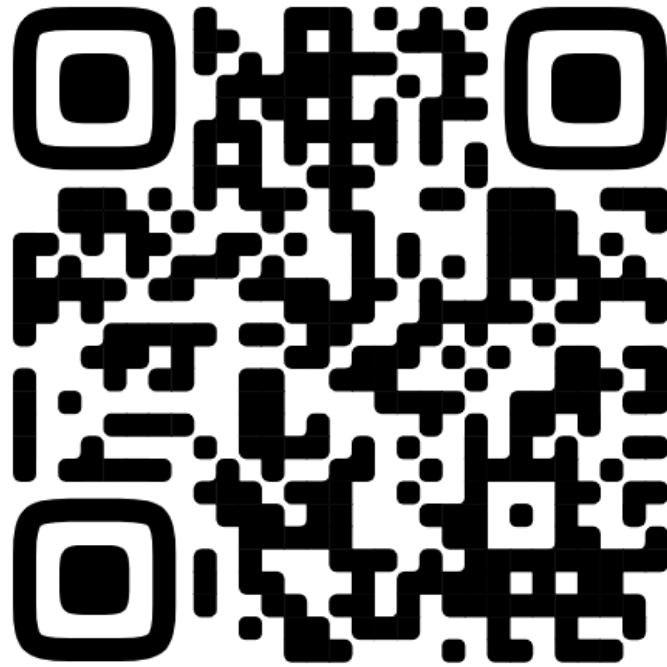


Ссылки

- Lena Voita – NLP Course: Seq2seq and Attention
 - https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html
- Jay Alammam – Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)
 - <https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention>
- Dive into Deep Learning – Attention Mechanisms
 - https://d2l.ai/chapter_attention-mechanisms/index.html

Доступ к общей папке

- <https://clck.ru/3EdrE3>



Transformer

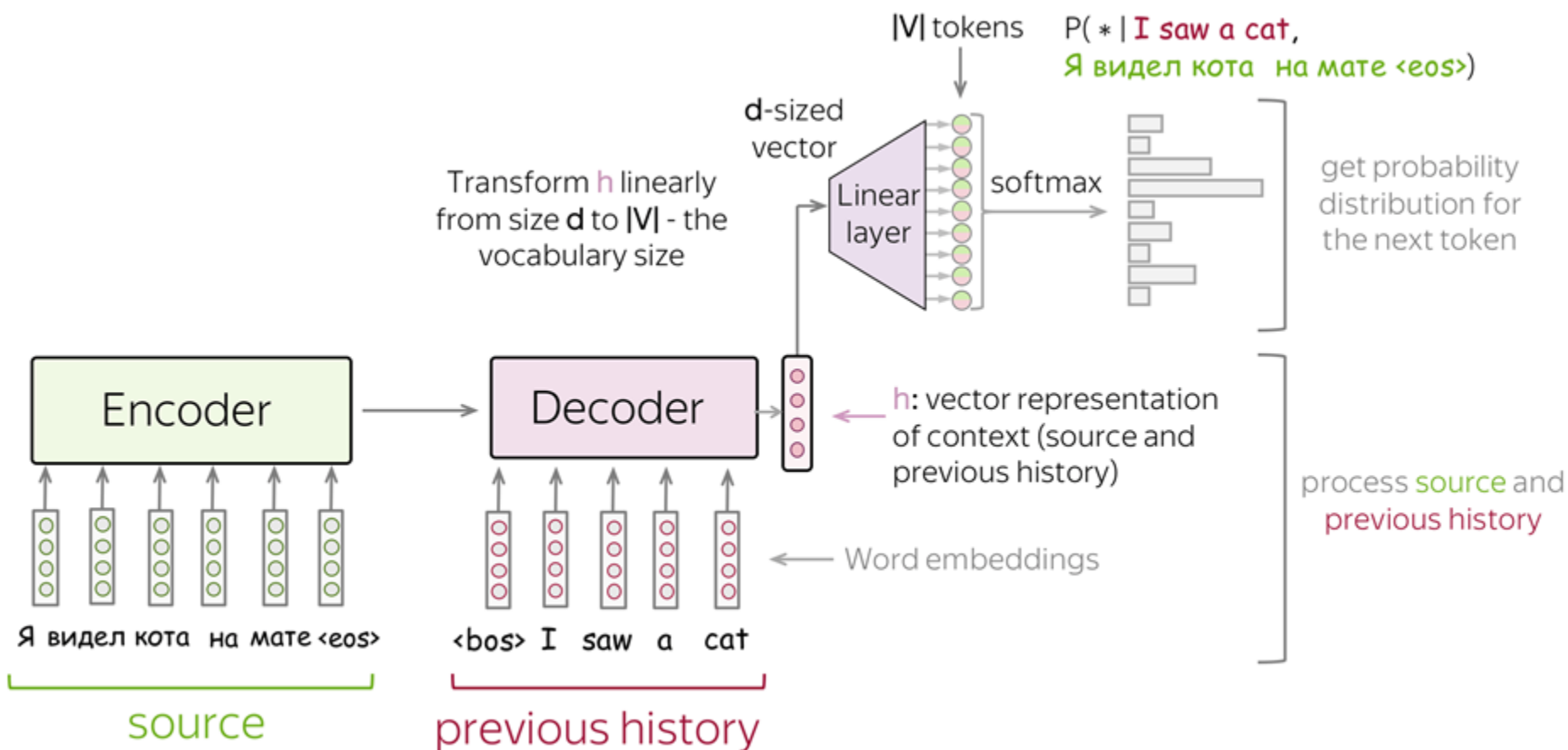
2024



Transformer

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin (Google). Attention Is All You Need // 31st Conference on Neural Information Processing Systems (NIPS), 2017
 - [Статья](#)
 - [arxiv.org](https://arxiv.org/abs/1706.03762): 12 июня 2017 года
- Нет рекуррентных и свёрточных блоков, только блоки внимания и прямого распространения

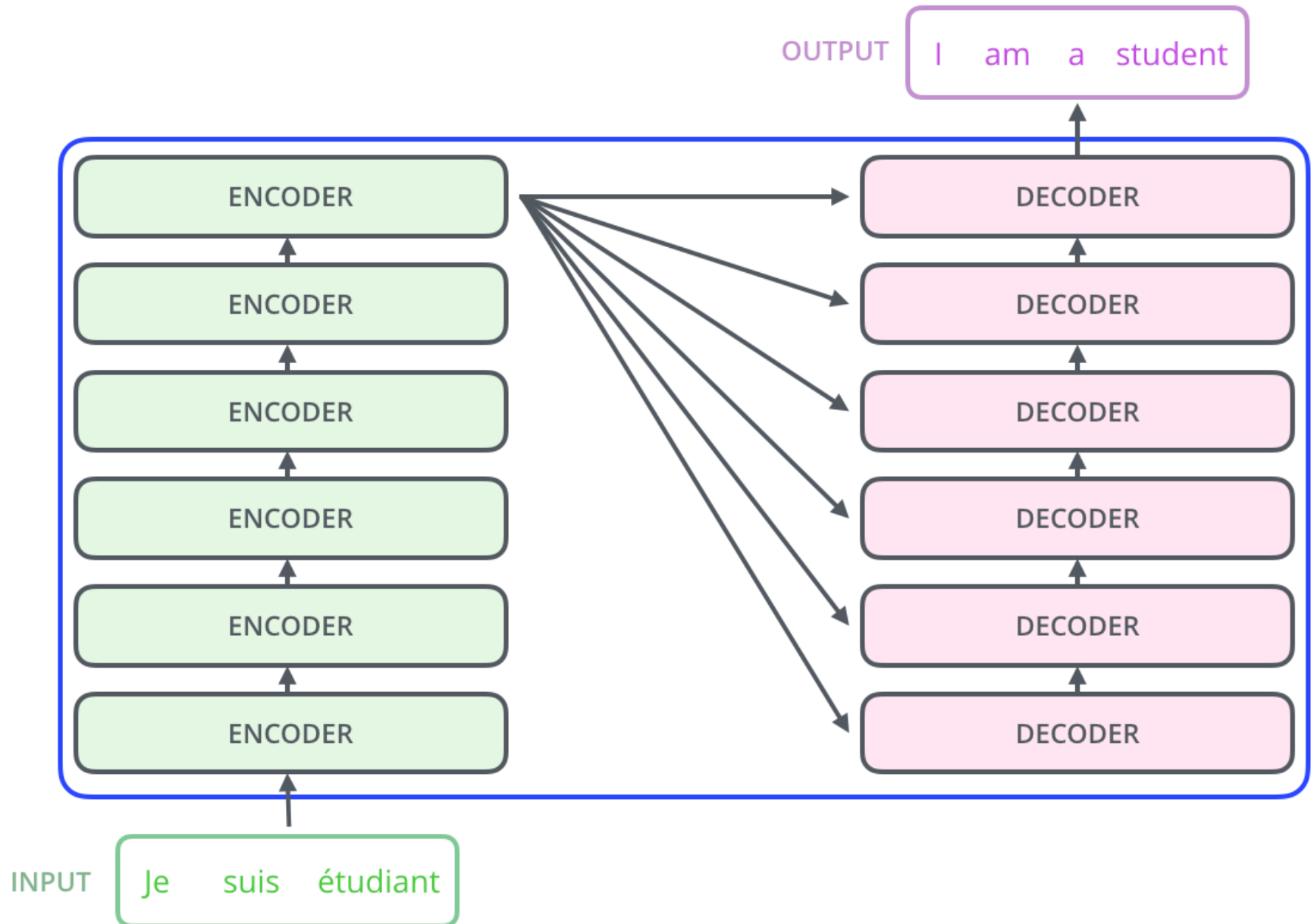
Seq2seq



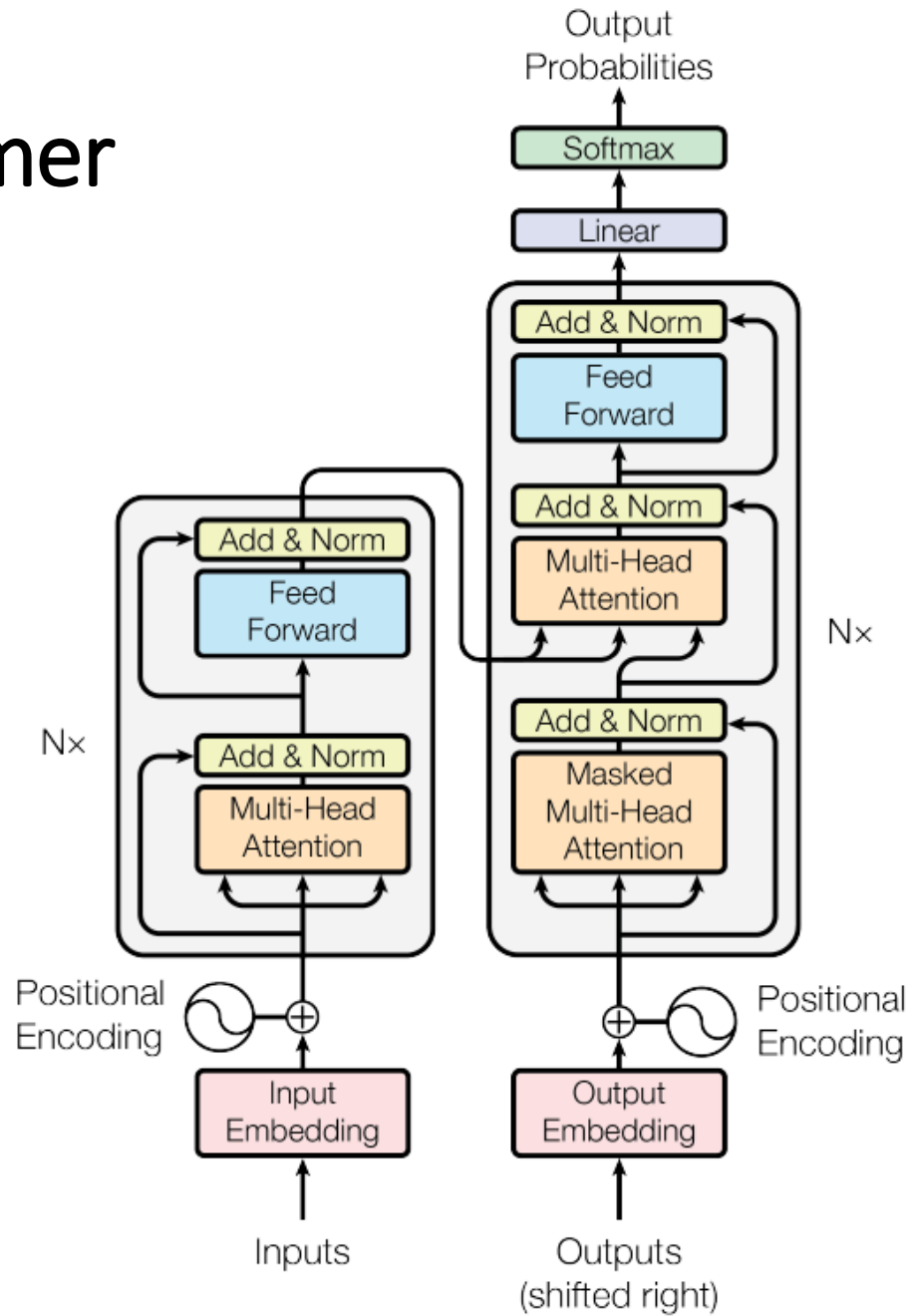
Transformer

	Seq2seq without attention	Seq2seq with attention	Transformer
processing within encoder	RNN/CNN	RNN/CNN	attention
processing within decoder	RNN/CNN	RNN/CNN	attention
decoder-encoder interaction	static fixed- sized vector	attention	attention

Transformer

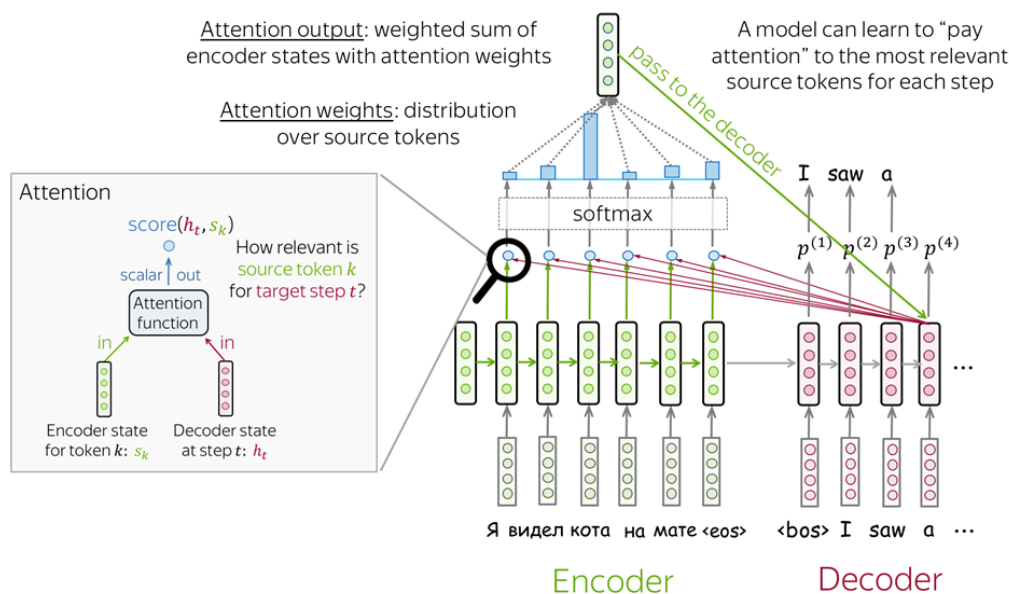


Transformer



Transformer: Self-Attention

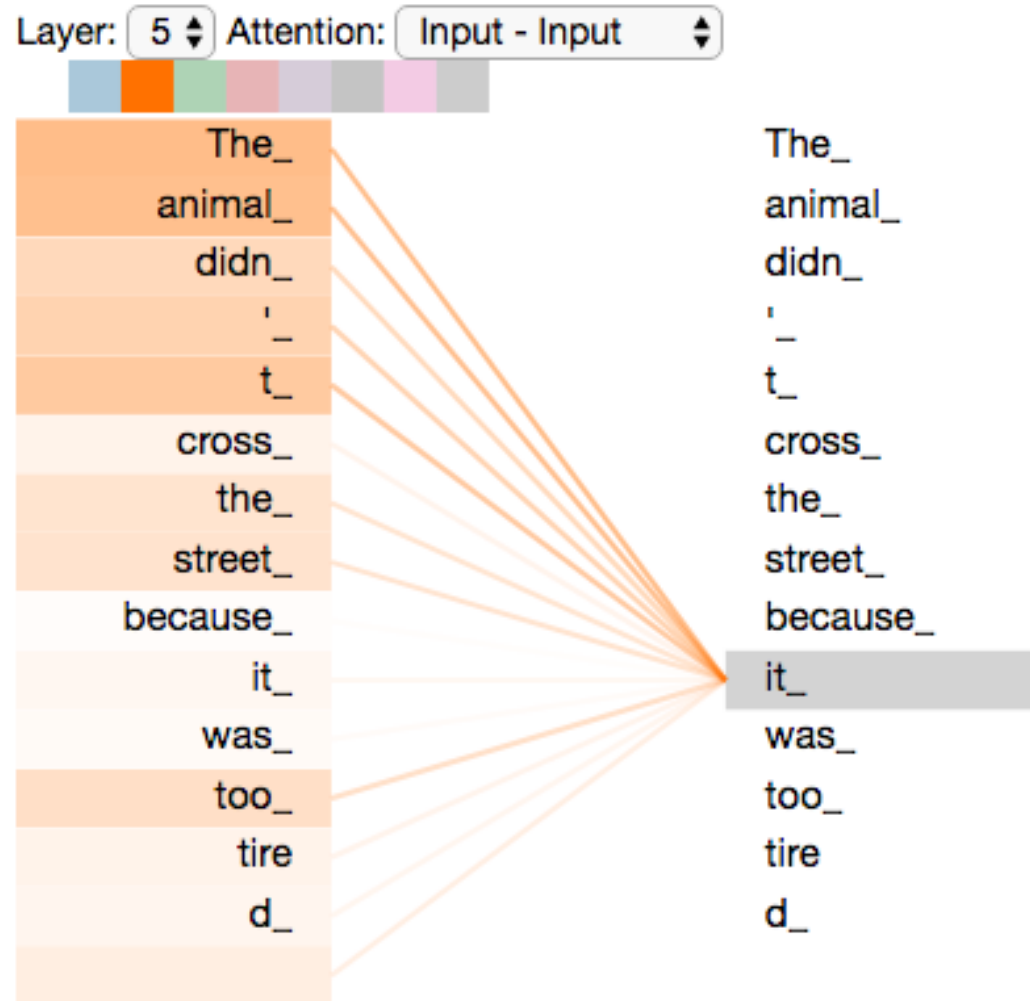
- Введен механизм внутреннего внимания (self-attention)
- Отличия от «обычного» внимания (cross-attention):
 - «Обычное» внимание между кодировщиком и декодировщиком направлено от текущего состояния декодировщика ко всем состояниям кодировщика



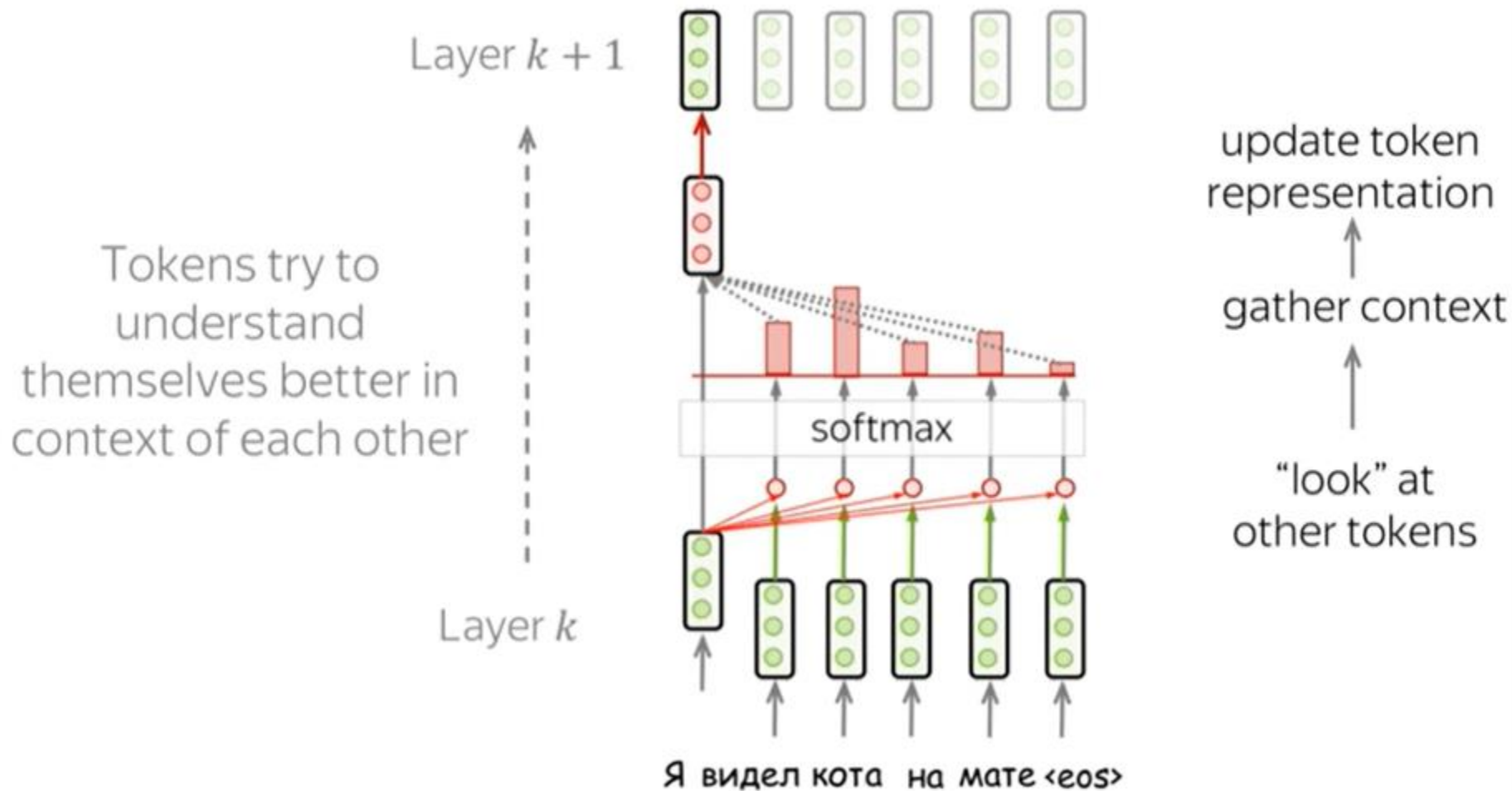
Transformer: Self-Attention

- Введен механизм внутреннего внимания (self-attention)
- Отличия от «обычного» внимания (cross-attention):
 - «Обычное» внимание между кодировщиком и декодировщиком направлено от текущего состояния декодировщика ко всем состояниям кодировщика
 - Внутреннее внимание направлено от каждого состояния из множества состояний (кодировщика или декодировщика) ко всем состояниям того же множества
 - После этого каждое состояние обновляется
 - Квадратичная сложность

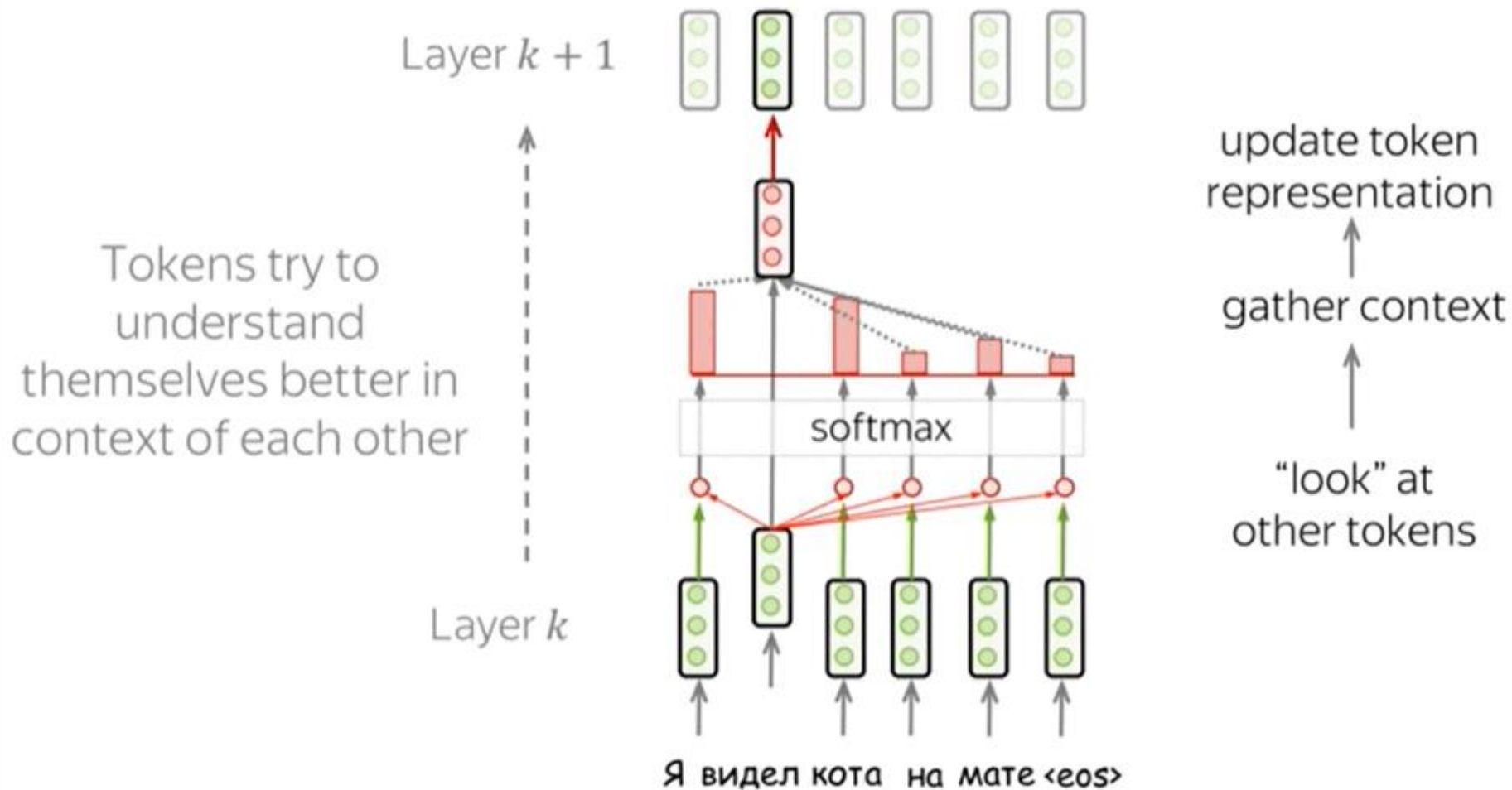
Transformer: Self-Attention



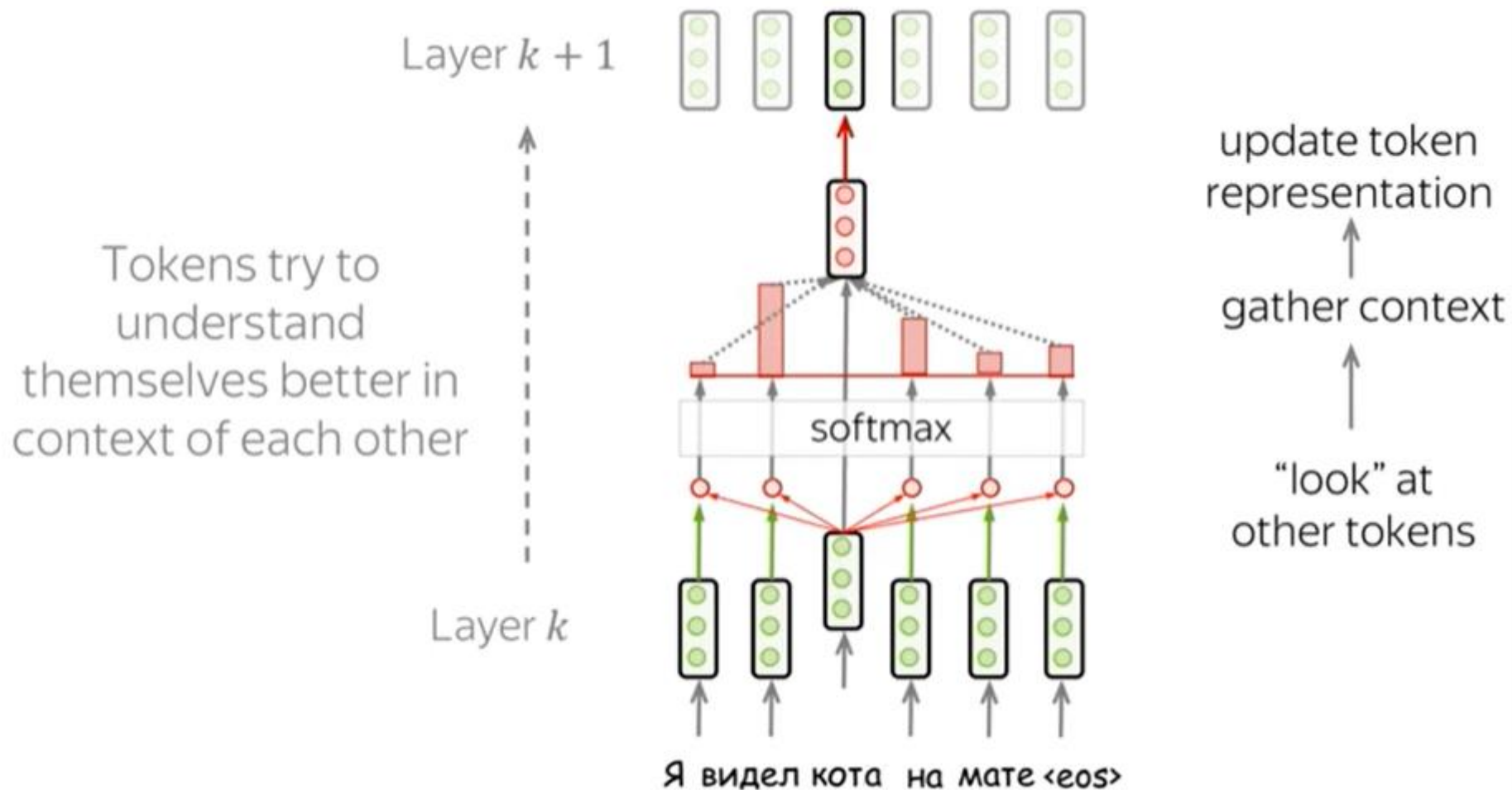
Transformer: Self-Attention



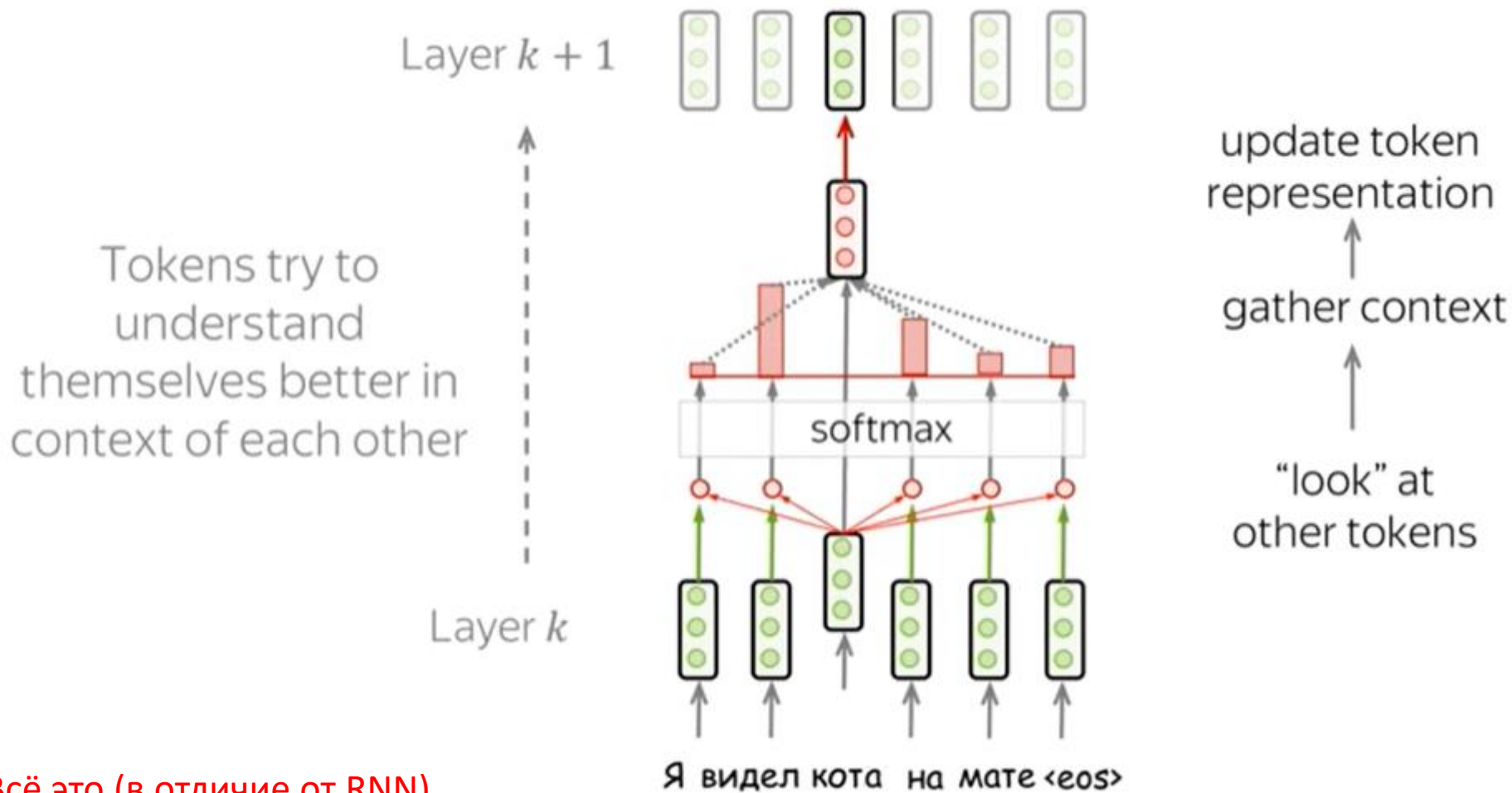
Transformer: Self-Attention



Transformer: Self-Attention



Transformer: Self-Attention



Всё это (в отличие от RNN)
происходит одновременно

Transformer: Query, Key, Value

- Внутреннее внимание реализуется в Трансформере при помощи *запросов* (queries), *ключей* (keys) и *значений* (values)
- Каждый токен имеет три представления:
 - *Query* – запрос на информацию
 - *Key* – наличие/отсутствие информации
 - *Value* – предоставление информации

Transformer: Query, Key, Value

Each vector receives three representations (“roles”)

$$\begin{bmatrix} W_Q \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{blue} \\ \text{blue} \\ \text{blue} \end{bmatrix}$$

Query: vector **from** which the attention is looking

“Hey there, do you have this information?”

$$\begin{bmatrix} W_K \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{yellow} \\ \text{yellow} \\ \text{yellow} \end{bmatrix}$$

Key: vector **at** which the query looks to compute weights

“Hi, I have this information – give me a large weight!”

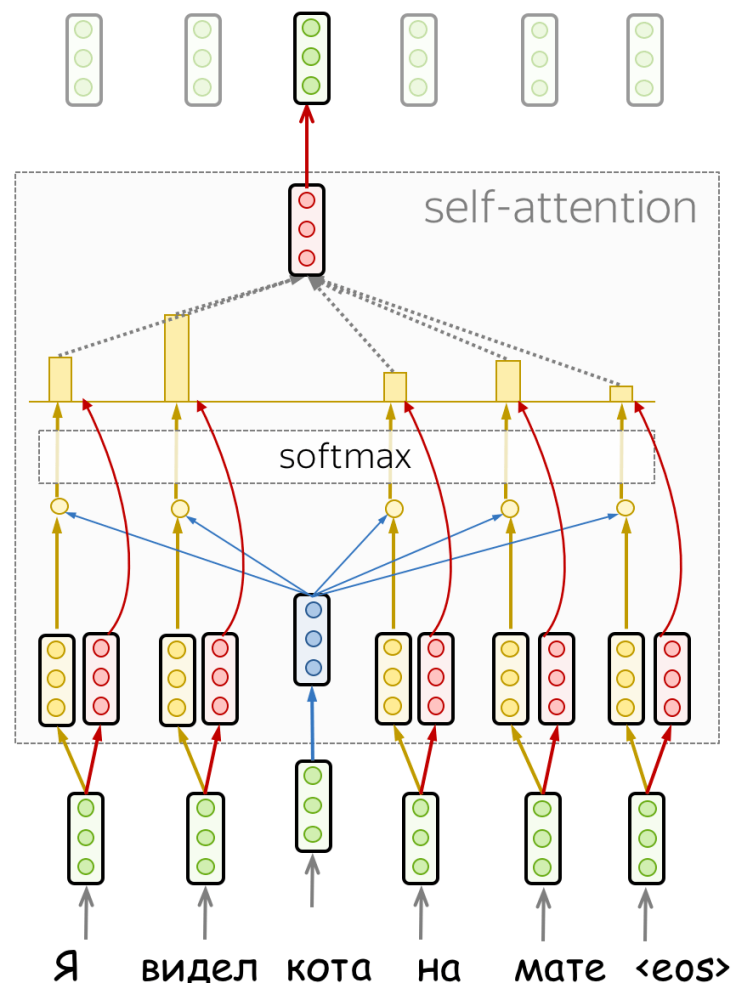
$$\begin{bmatrix} W_V \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{red} \\ \text{red} \\ \text{red} \end{bmatrix}$$

Value: their weighted sum is attention output

“Here’s the information I have!”

$$\underset{\substack{\text{from} \\ \text{to}}}{\text{Attention}}(q, k, v) = \overbrace{\text{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right)}^{\text{Attention weights}} v$$

vector dimensionality of K, V



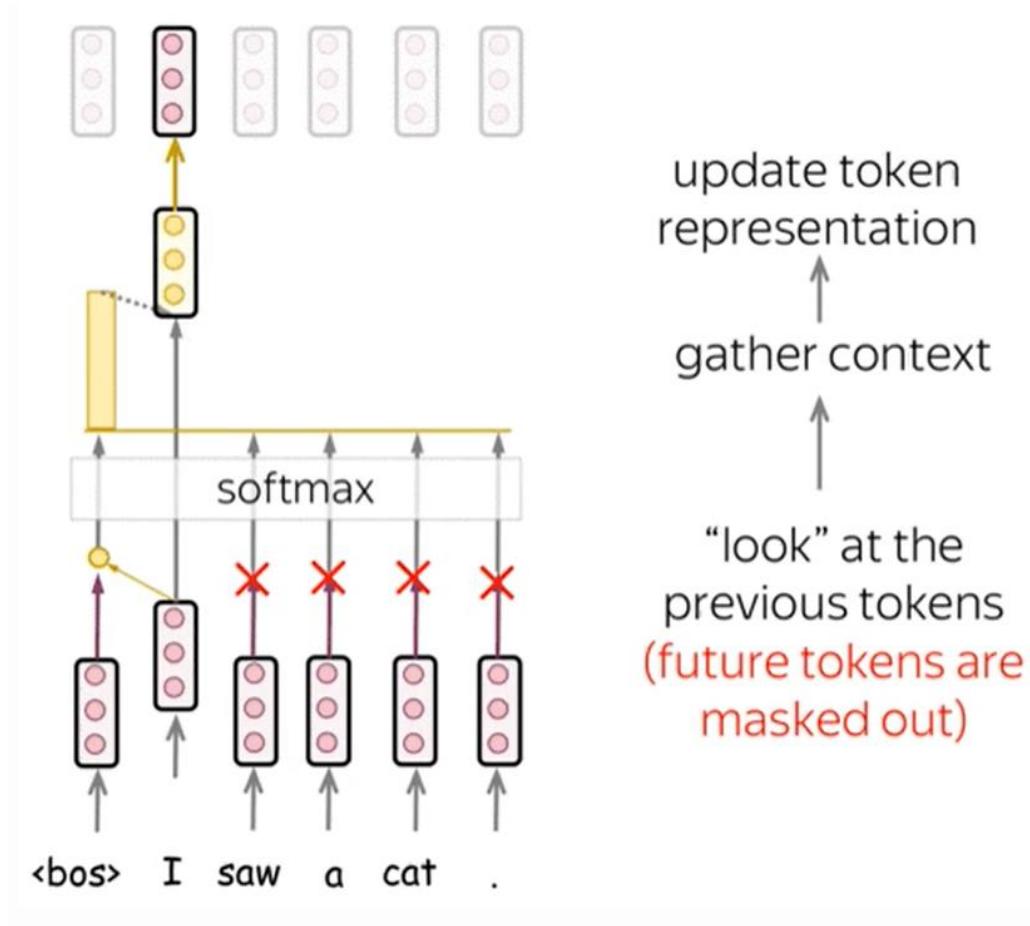
Transformer: Self-Attention

- Таким образом, векторное представление слова (токена) будет контекстно-зависимым, в отличие от word2vec и GloVe
- В исходной статье такой способ вычисления внимания называется «Scaled Dot-Product Attention»

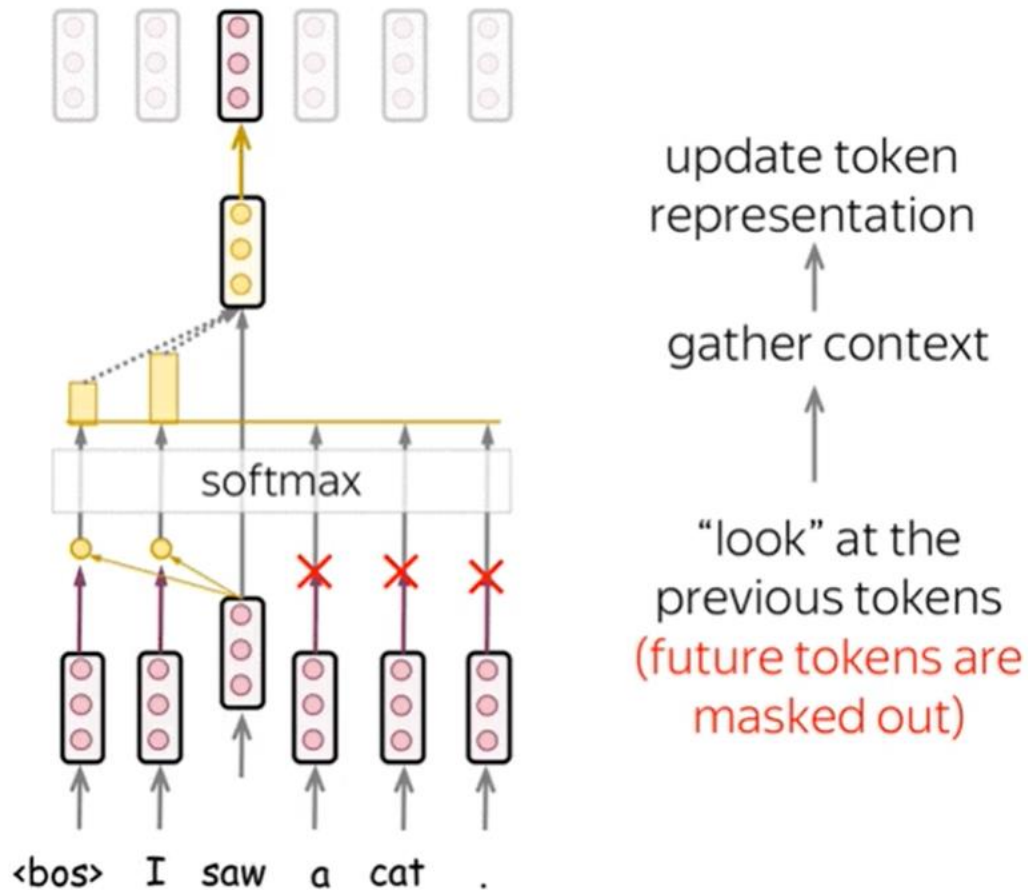
Transformer: Masked Self-Attention

- В декодировщике все токены, следующие за текущим, маскированы:
 - в процессе генерации (инференса) они неизвестны
 - в процессе обучения они известны, но не должны влиять на декодировщик

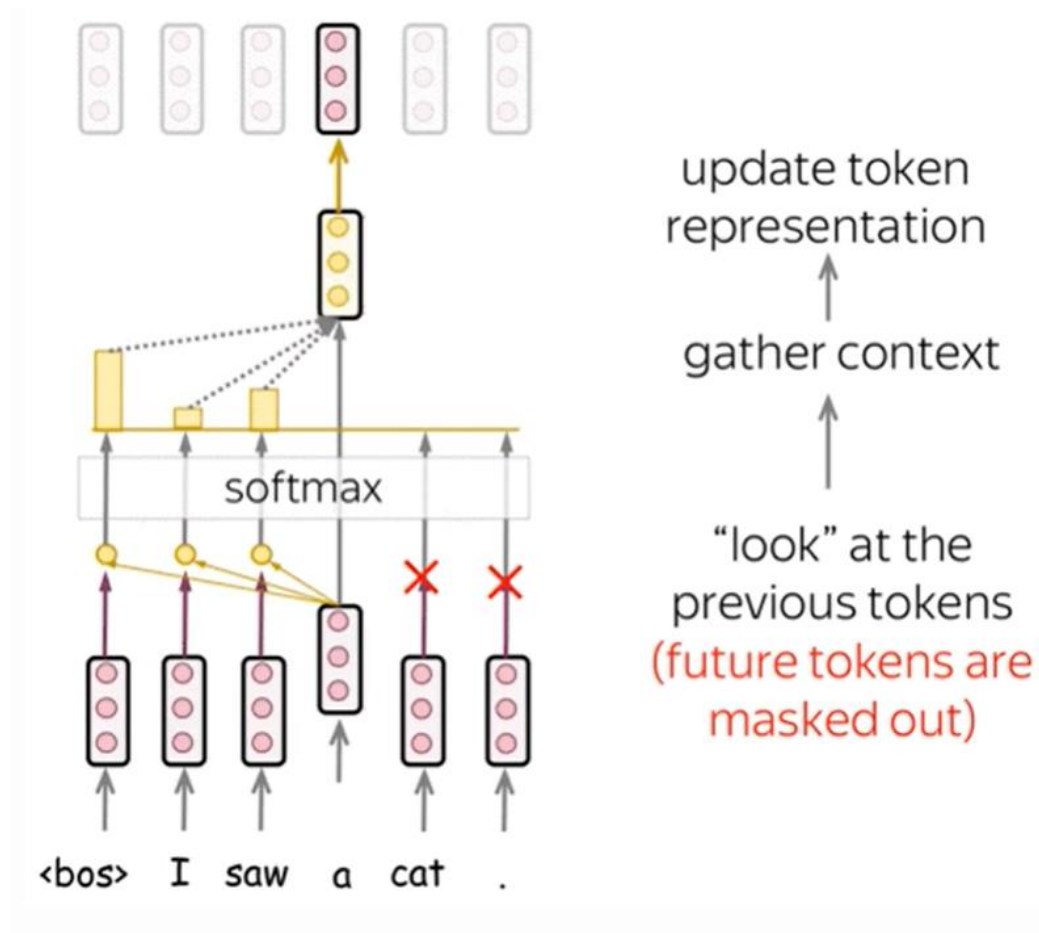
Transformer: Masked Self-Attention



Transformer: Masked Self-Attention



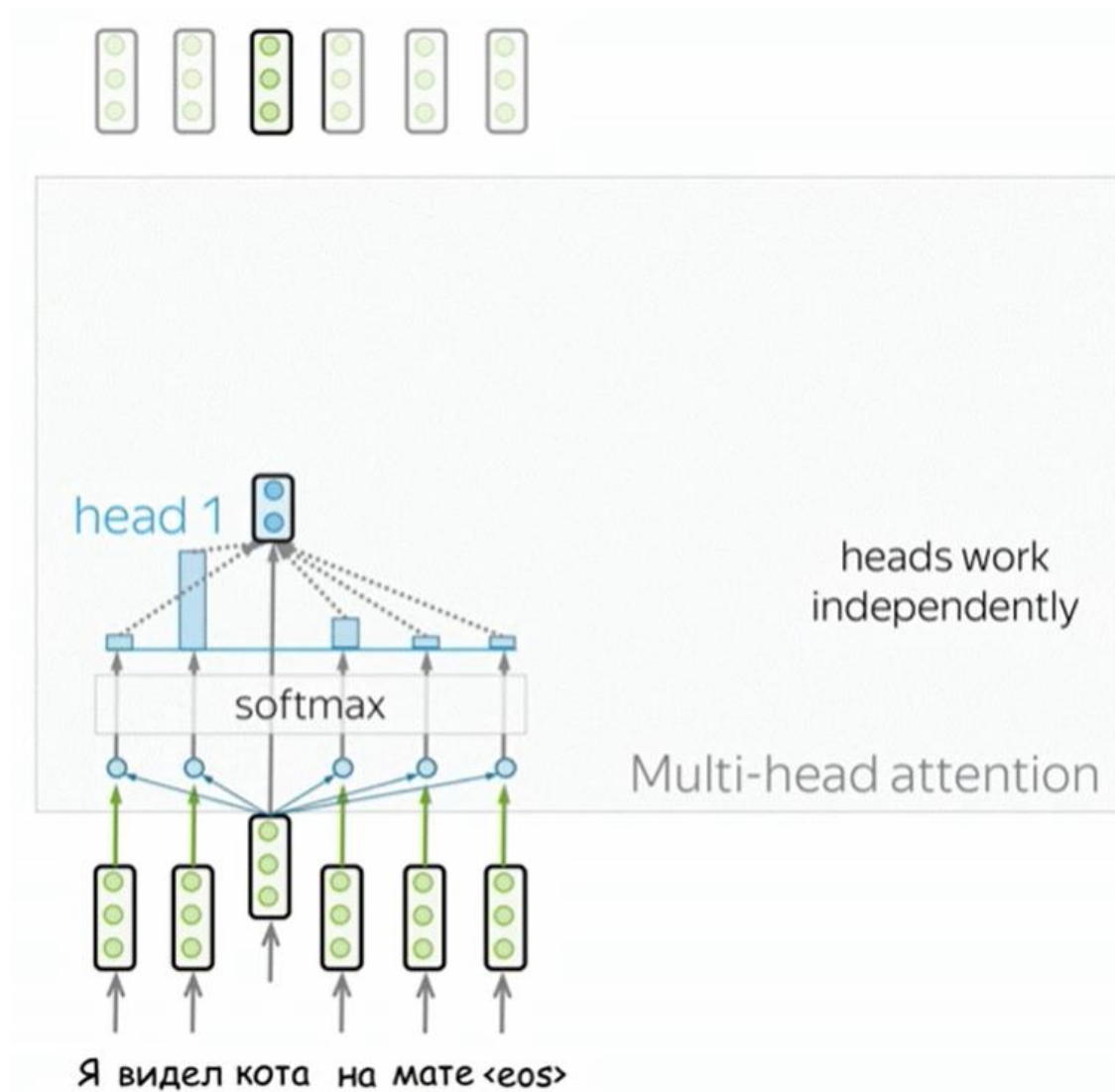
Transformer: Masked Self-Attention



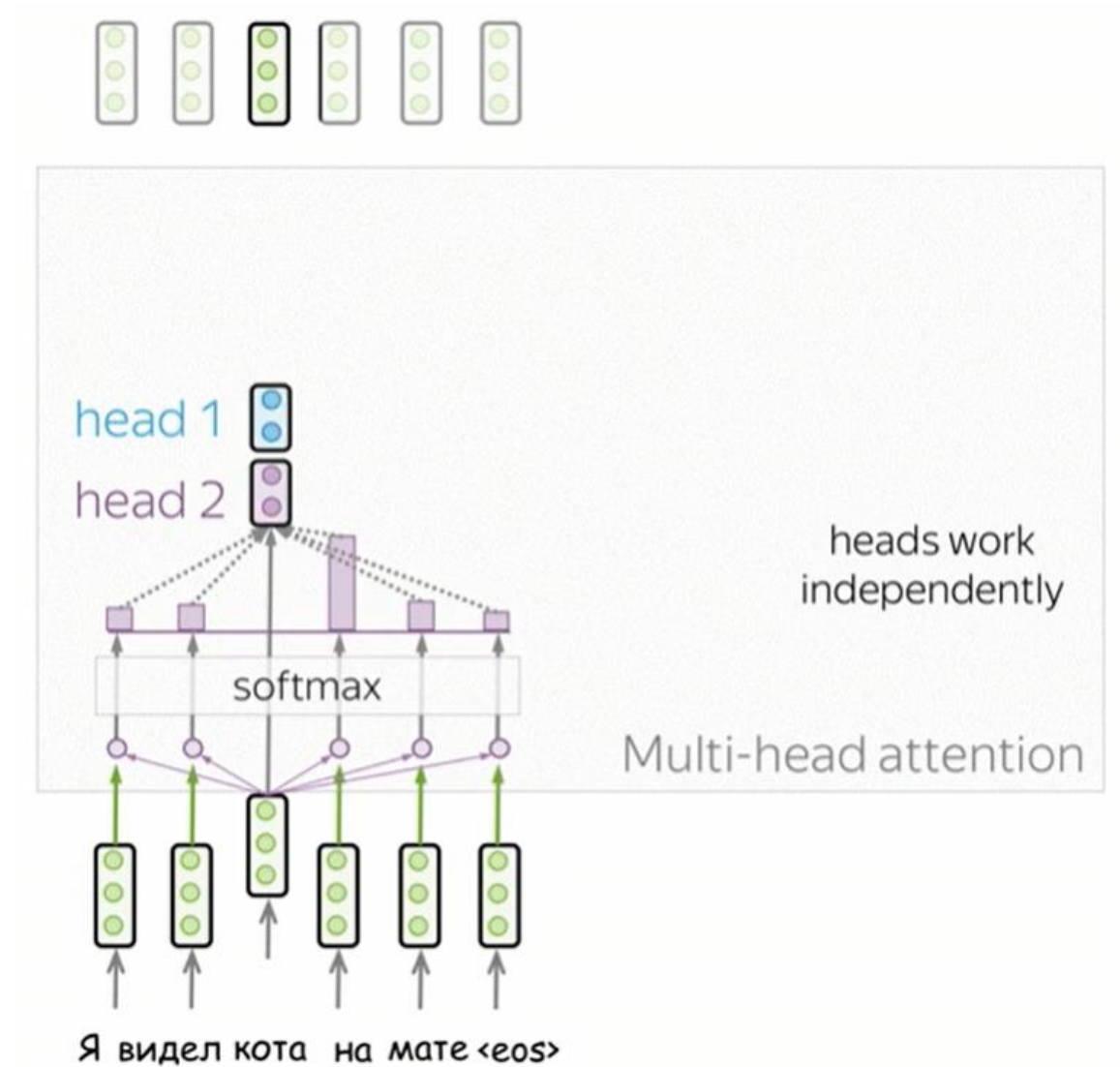
Transformer: Multi-Head Attention

- Одно и то же слово может быть связано с разными аспектами в процессе понимания смысла предложения – морфологическими, синтаксическими, семантическими
- Поэтому в Трансформере внимание распределено по «головам» (heads) – отдельным матрицам, которые работают независимо
- Примеры ролей «голов» [[Voita et al., 2019](#)]:
 - позиционные – внимание на двух-трёх предыдущих/следующих токенах
 - синтаксические – подлежащее-сказуемое, глагол-дополнение
 - редкие токены

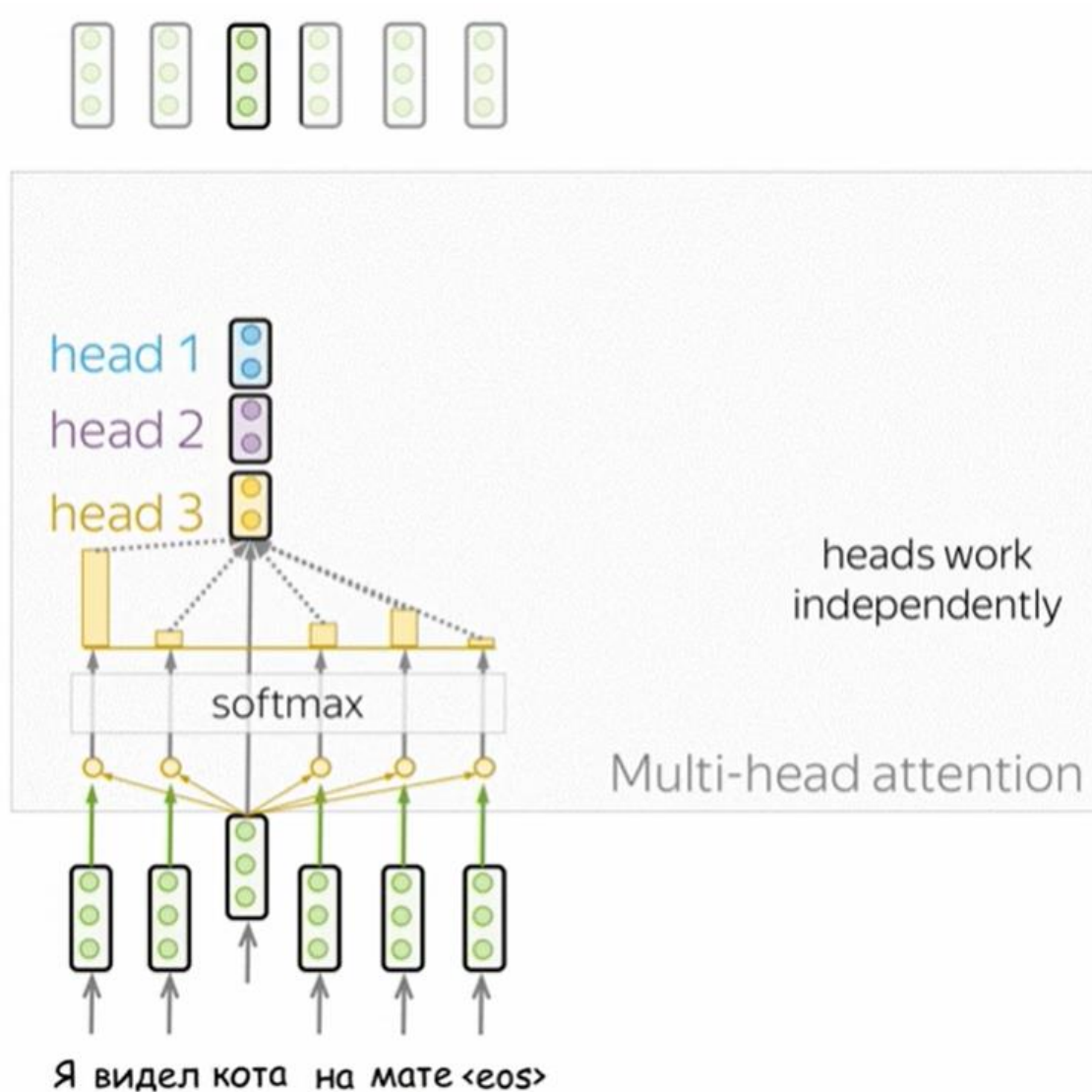
Transformer: Multi-Head Attention



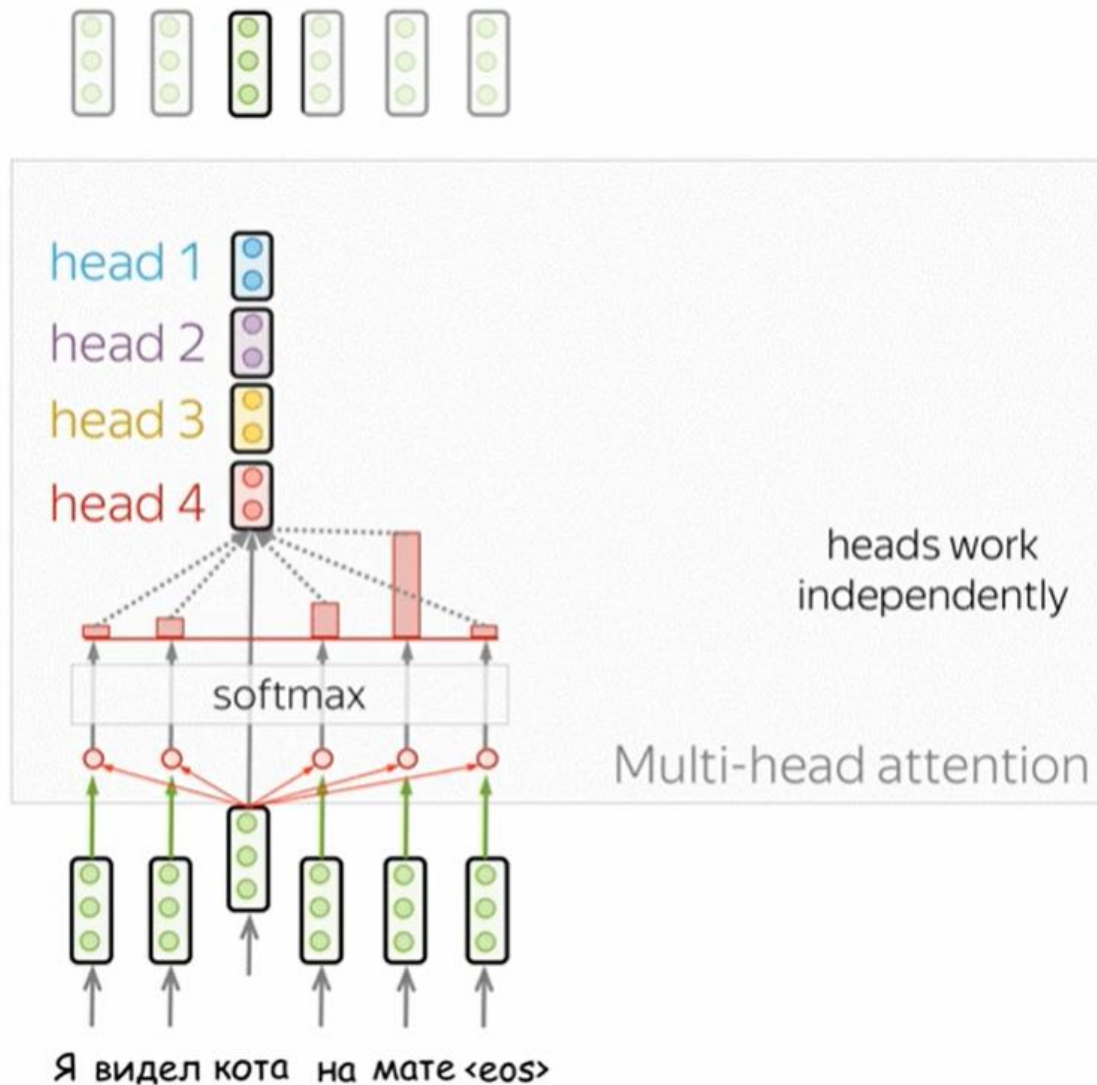
Transformer: Multi-Head Attention



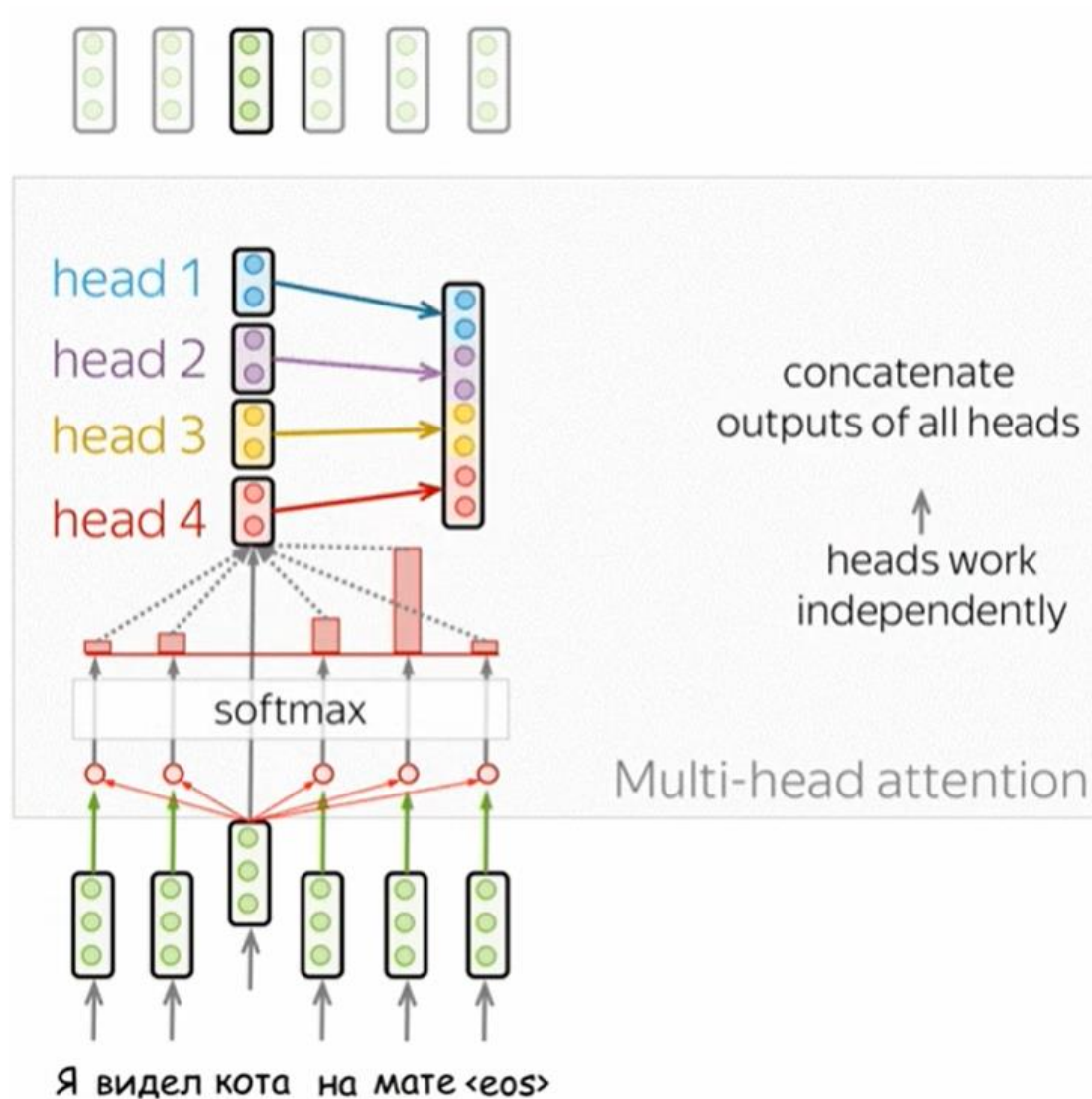
Transformer: Multi-Head Attention



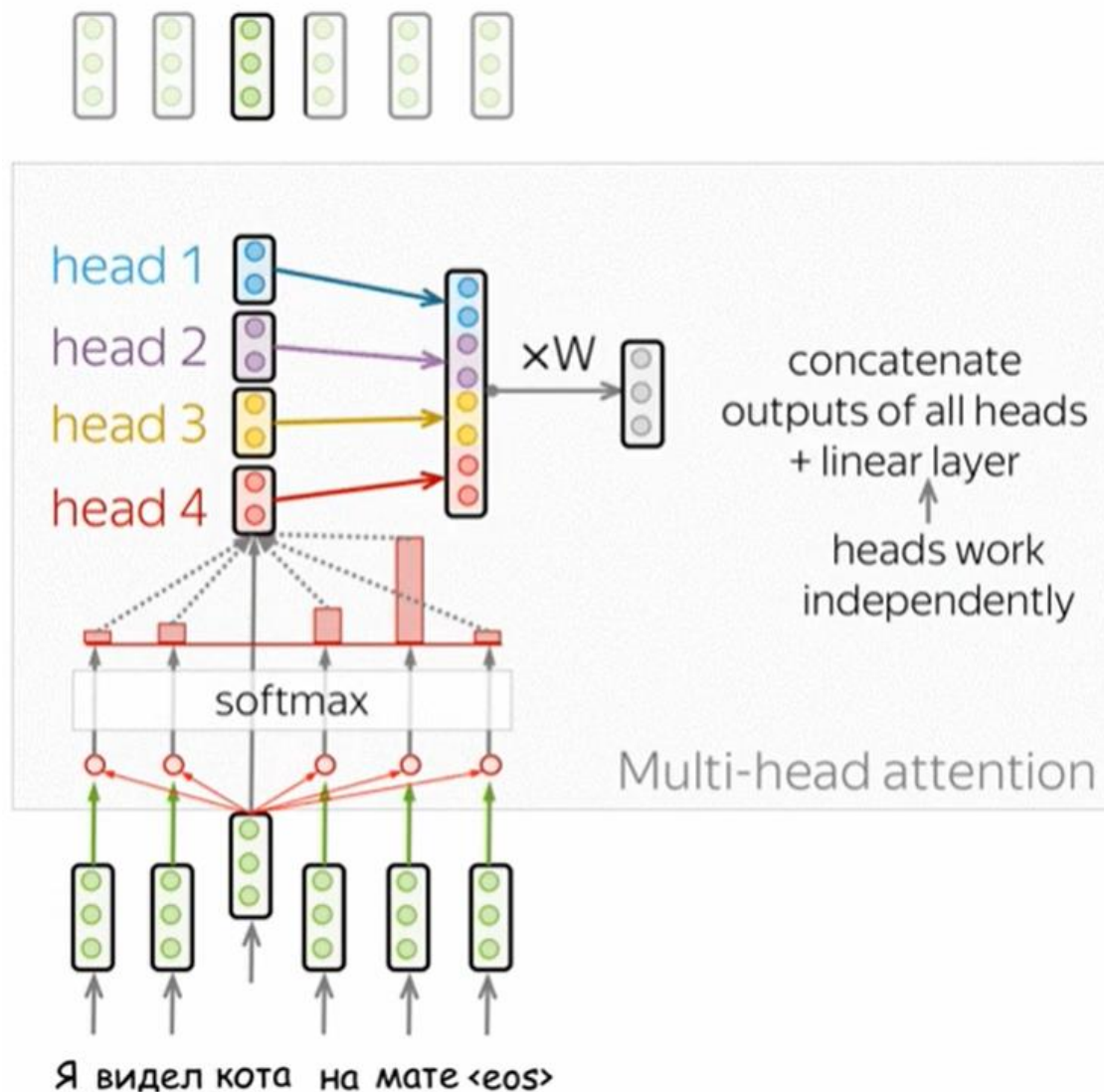
Transformer: Multi-Head Attention



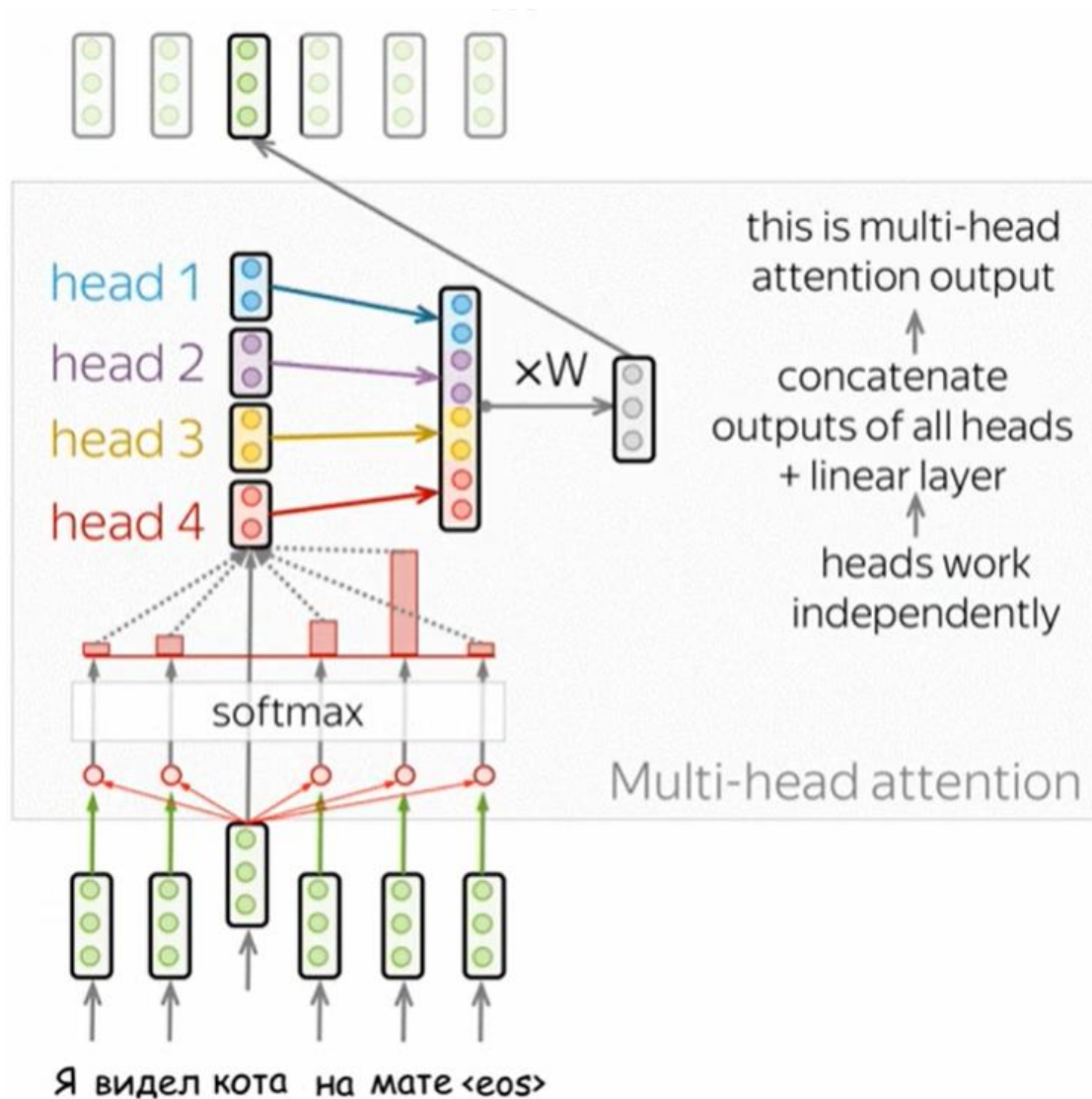
Transformer: Multi-Head Attention



Transformer: Multi-Head Attention



Transformer: Multi-Head Attention

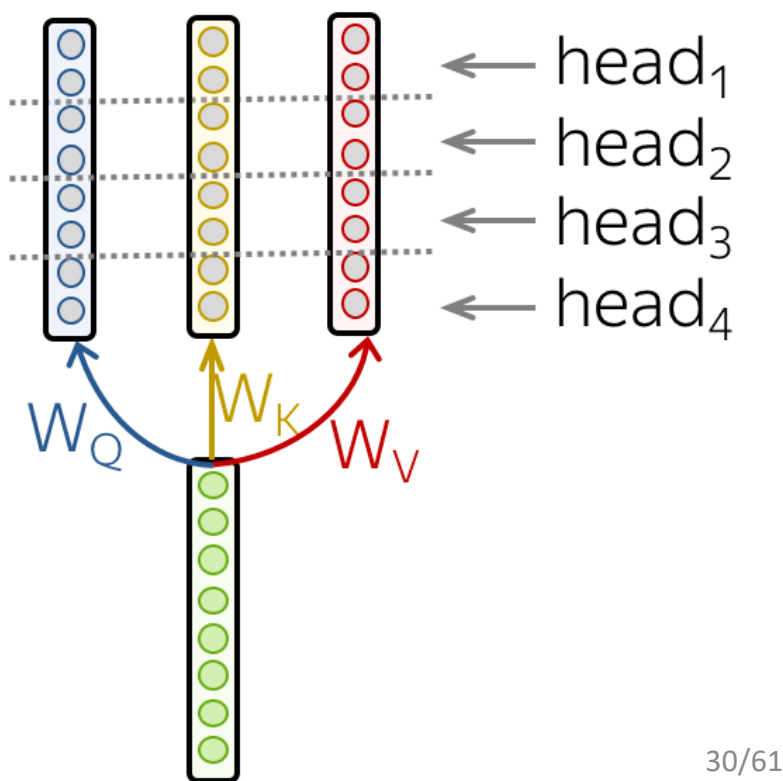


Transformer: Multi-Head Attention

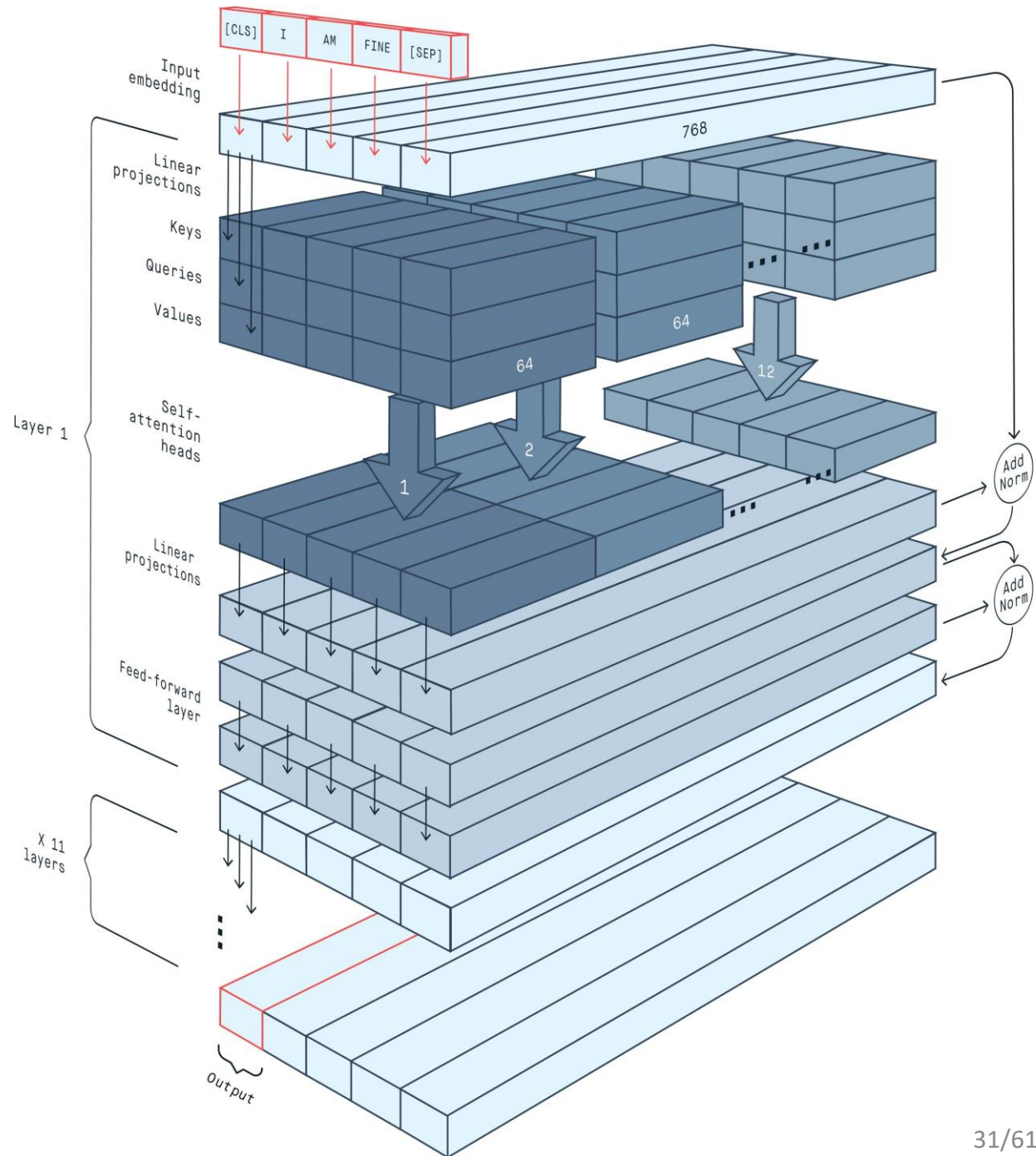
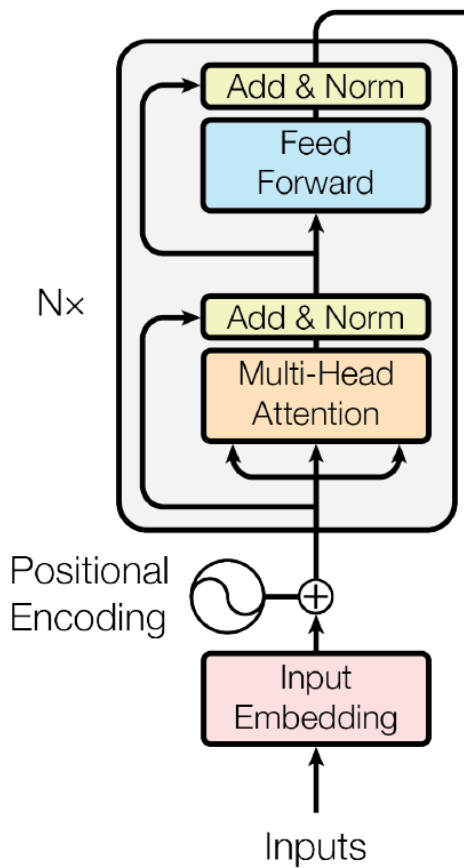
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_n)W_o,$$

где $\text{head}_i = \text{Attention}(QW_Q^i, KW_K^i, VW_V^i)$

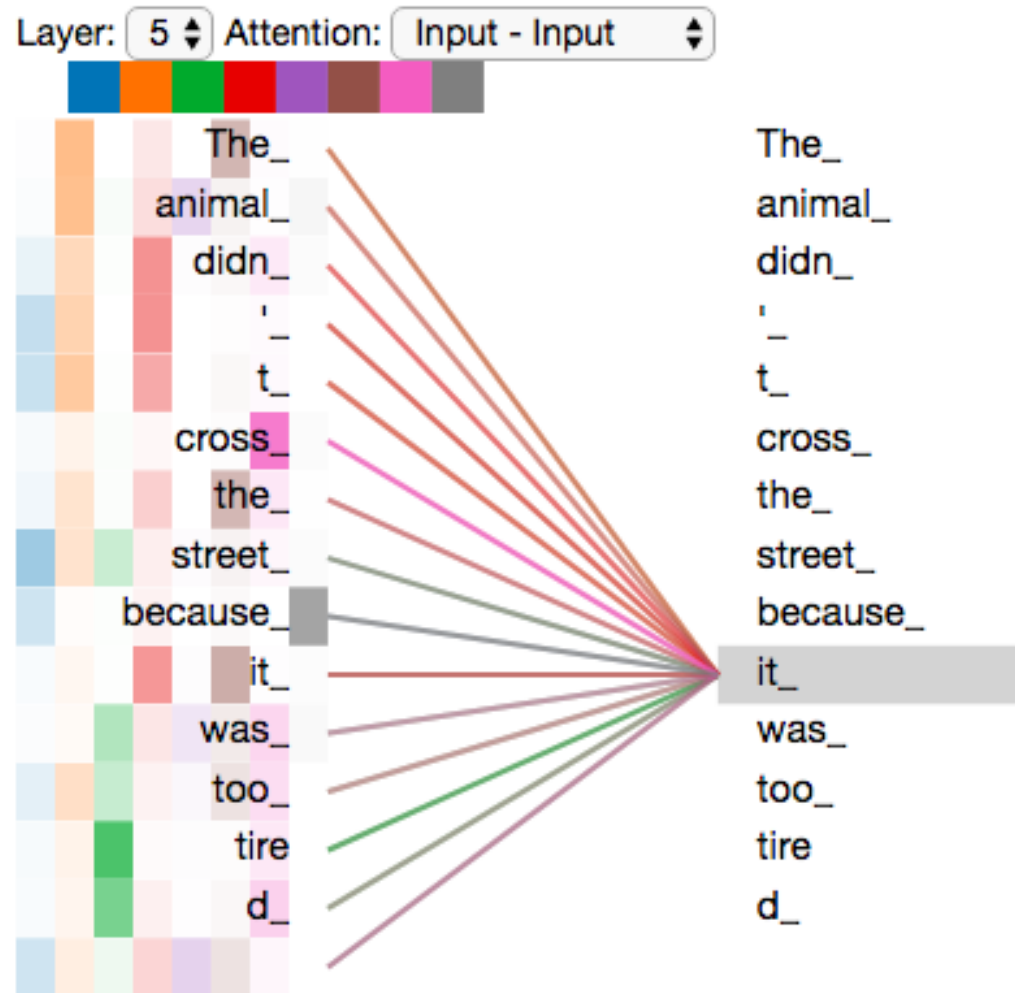
Split equally
into number of
heads parts



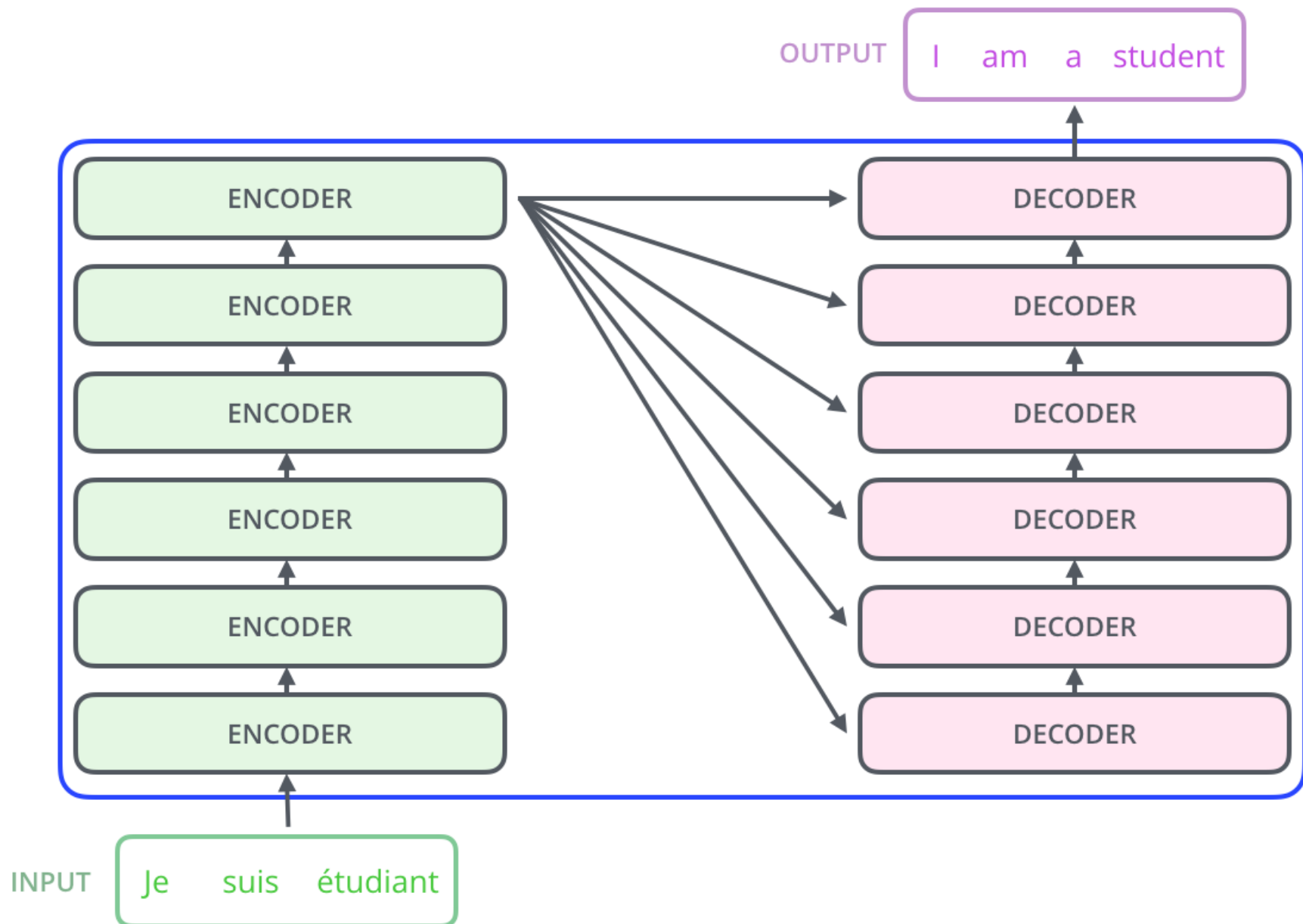
Transformer: Multi-Head Attention



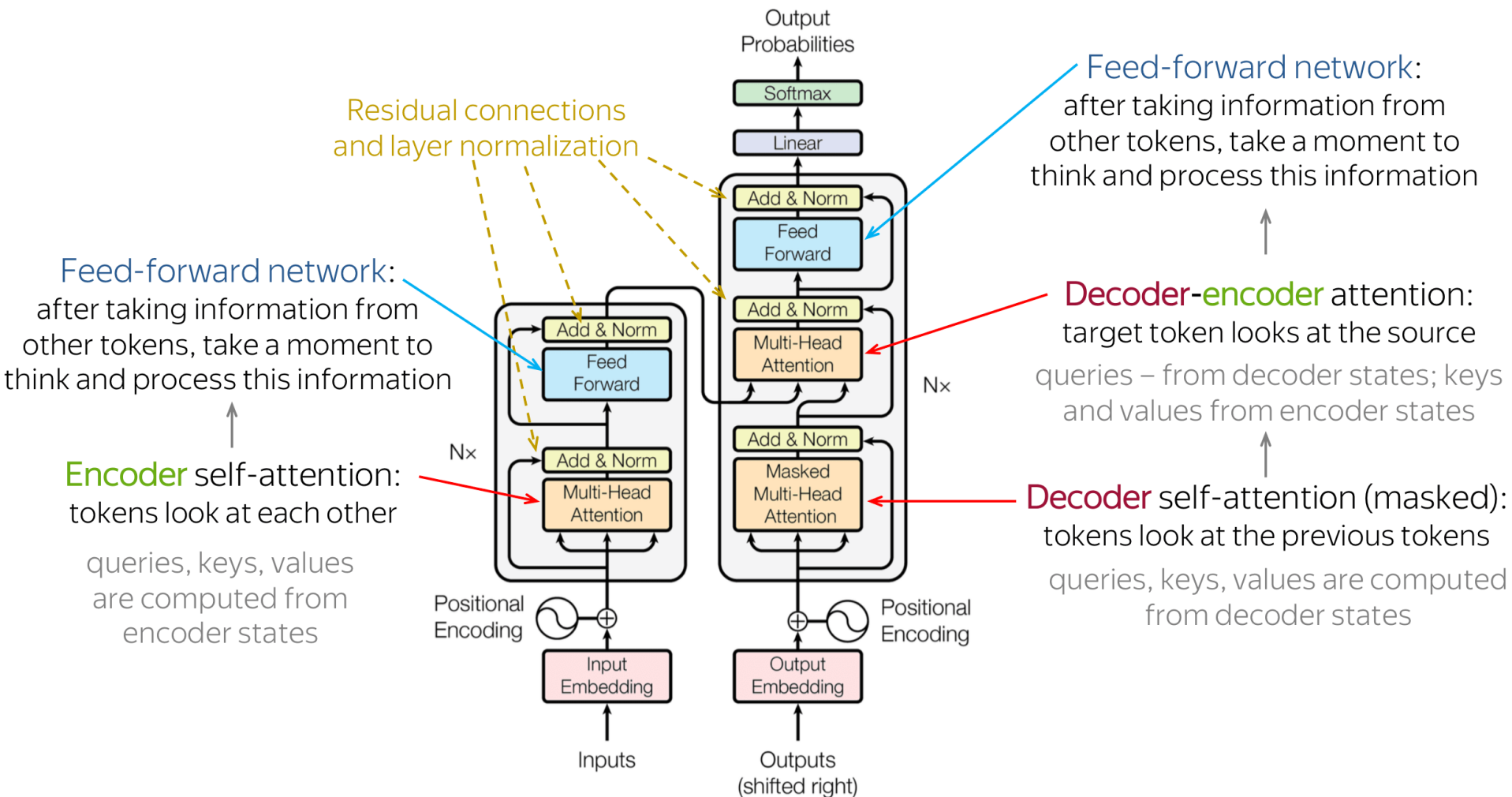
Transformer: Multi-Head Attention



Transformer: Model Architecture



Transformer: Model Architecture

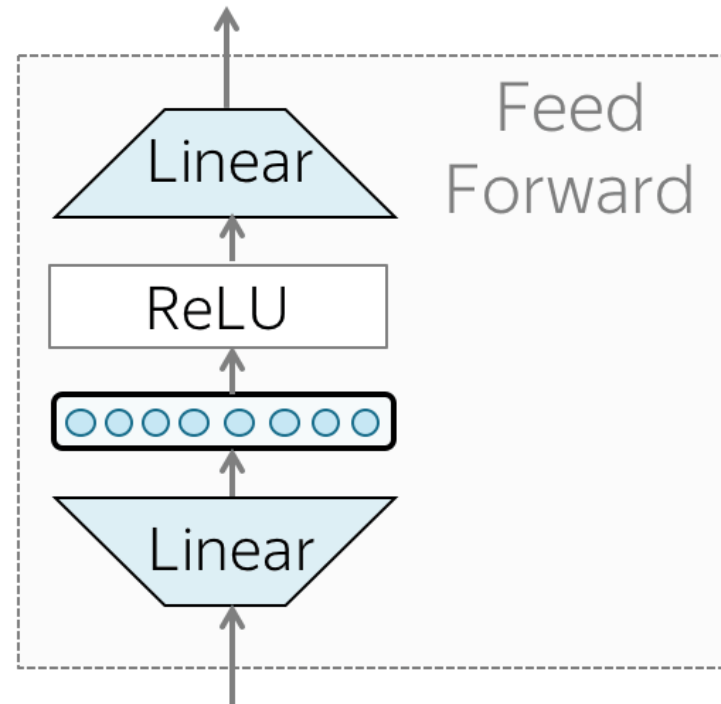
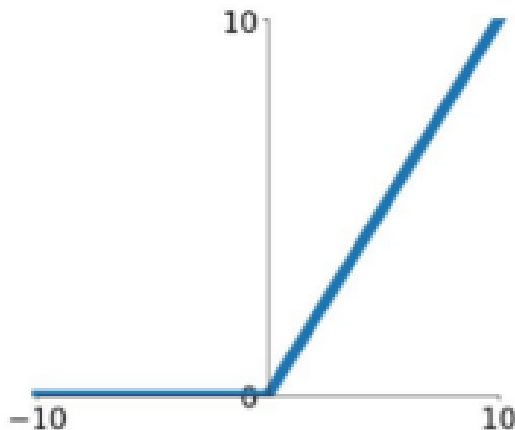


Transformer: Feed-forward blocks

- Двухслойный персептрон: формирование новых представлений токенов на основе комбинации предыдущих представлений (в том числе от разных голов)

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

$$ReLU(z) = \max(0, z)$$



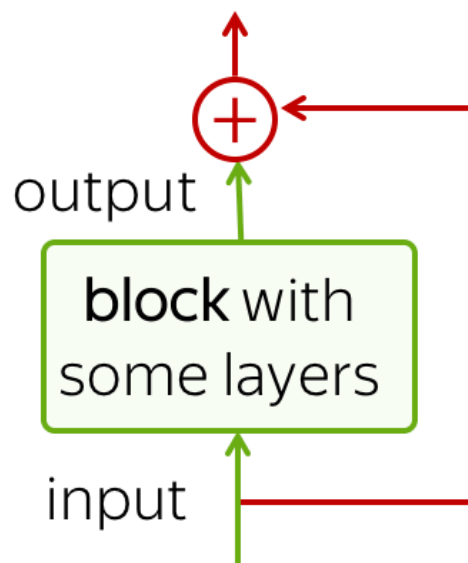
Transformer: Residual connections

- Проблема: затухание градиентов при обучении (vanishing gradients) – если градиент $\rightarrow 0$, то на предыдущем слое изменения весов не происходит
 - Чем больше слоев, тем сильнее
- Решение: остаточные связи (residual connections)

$$y = x + f(x)$$

$$f(x) = y - x$$

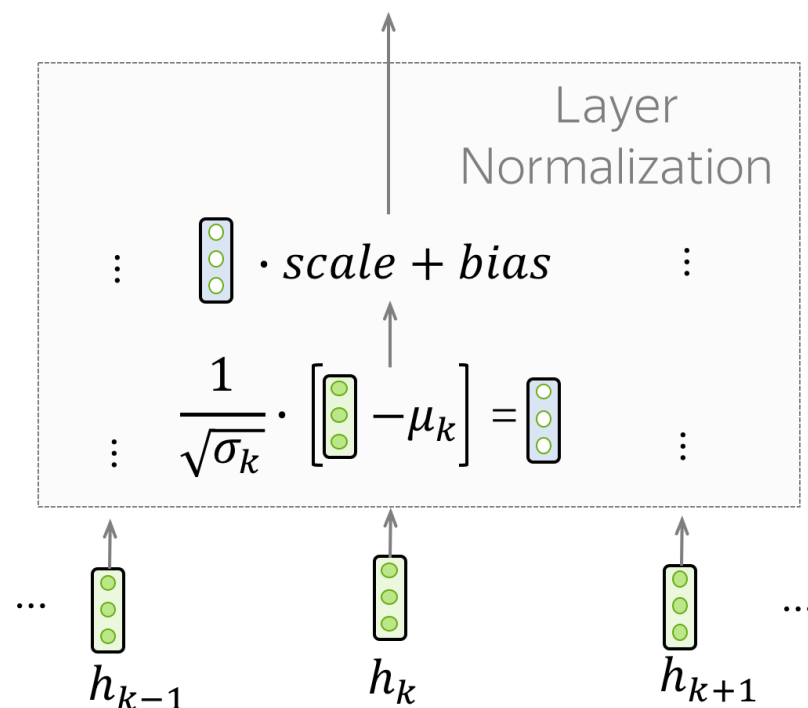
$$\frac{dy}{dx} = 1 + \frac{df(x)}{dx}$$



Residual connection:
add a block's input to
its output

Transformer: Layer Normalization

- Стандартная нормализация – используется для приведения векторов всех токенов к нормальному распределению
- μ_k, σ_k – среднее и стандартное отклонение
- $scale, bias$ – обучаемые параметры слоя нормализации
- [torch.nn.LayerNorm](#)
- [\[Ba et al. – Layer Normalization, 2016\]](#)



Transformer: Positional encoding

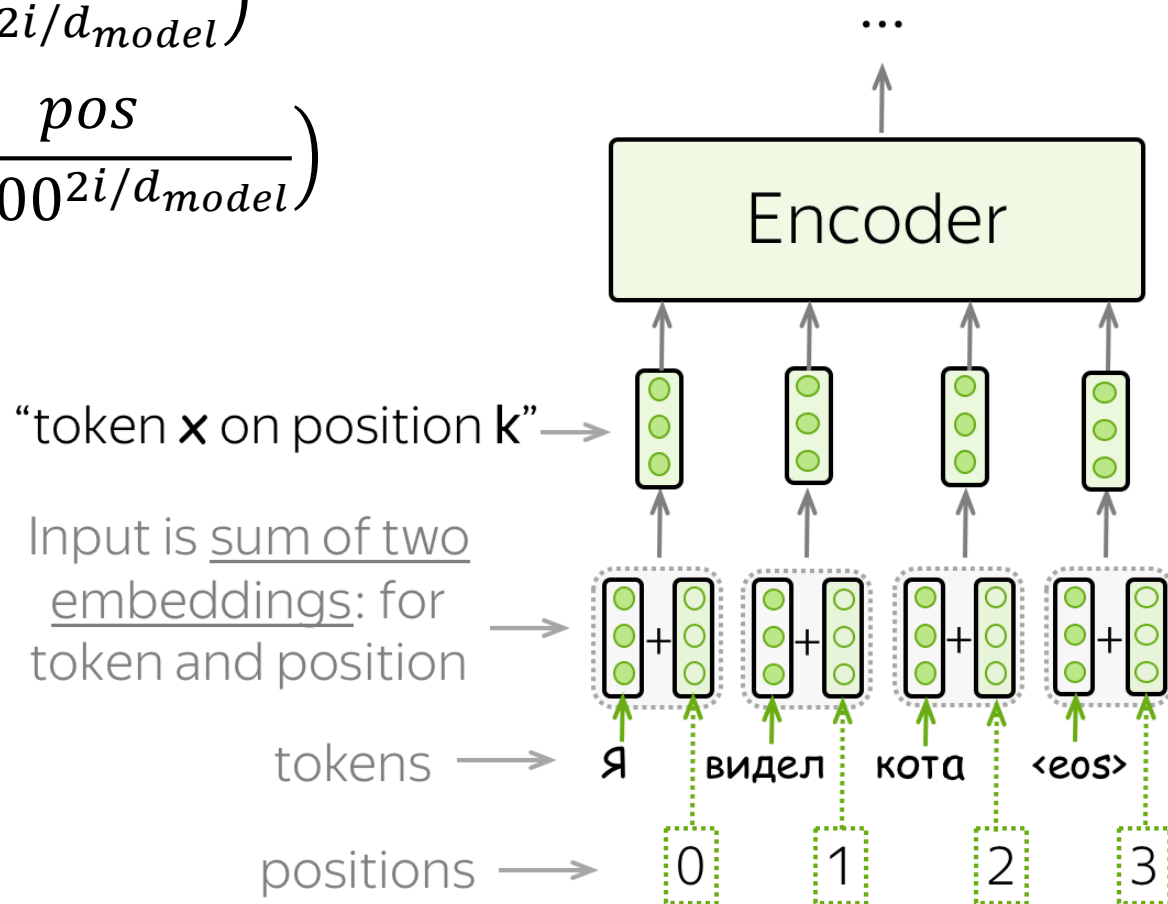
$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

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

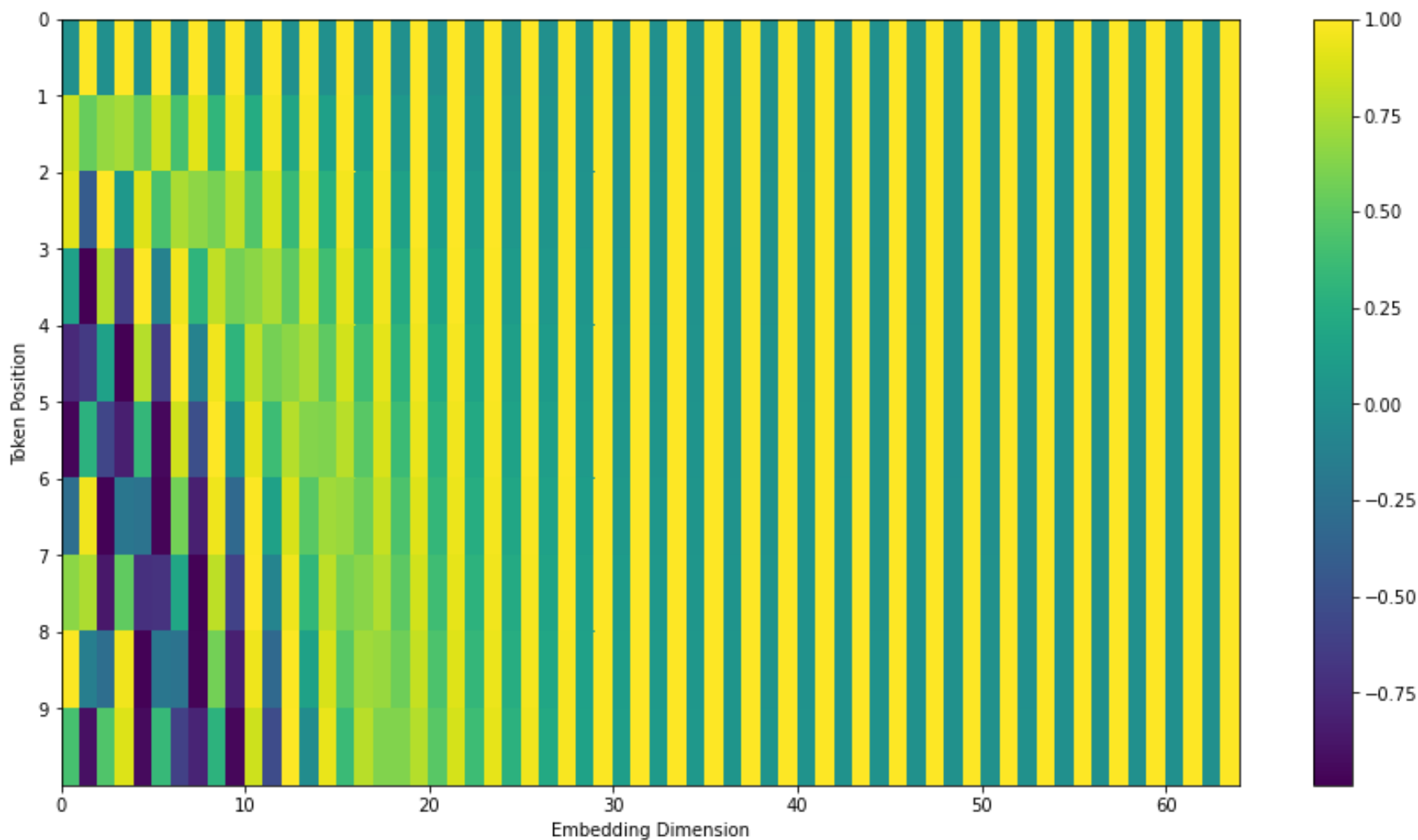
pos – позиция,

i – номер
размерности

Позиционные
эмбеддинги –
фиксированные
(не обучаются)



Transformer: Positional encoding



Transformer: Final Linear and Softmax Layers

Which word in our vocabulary
is associated with this index?

Get the index of the cell
with the highest value
(argmax)

am

5

log_probs



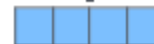
Softmax

logits

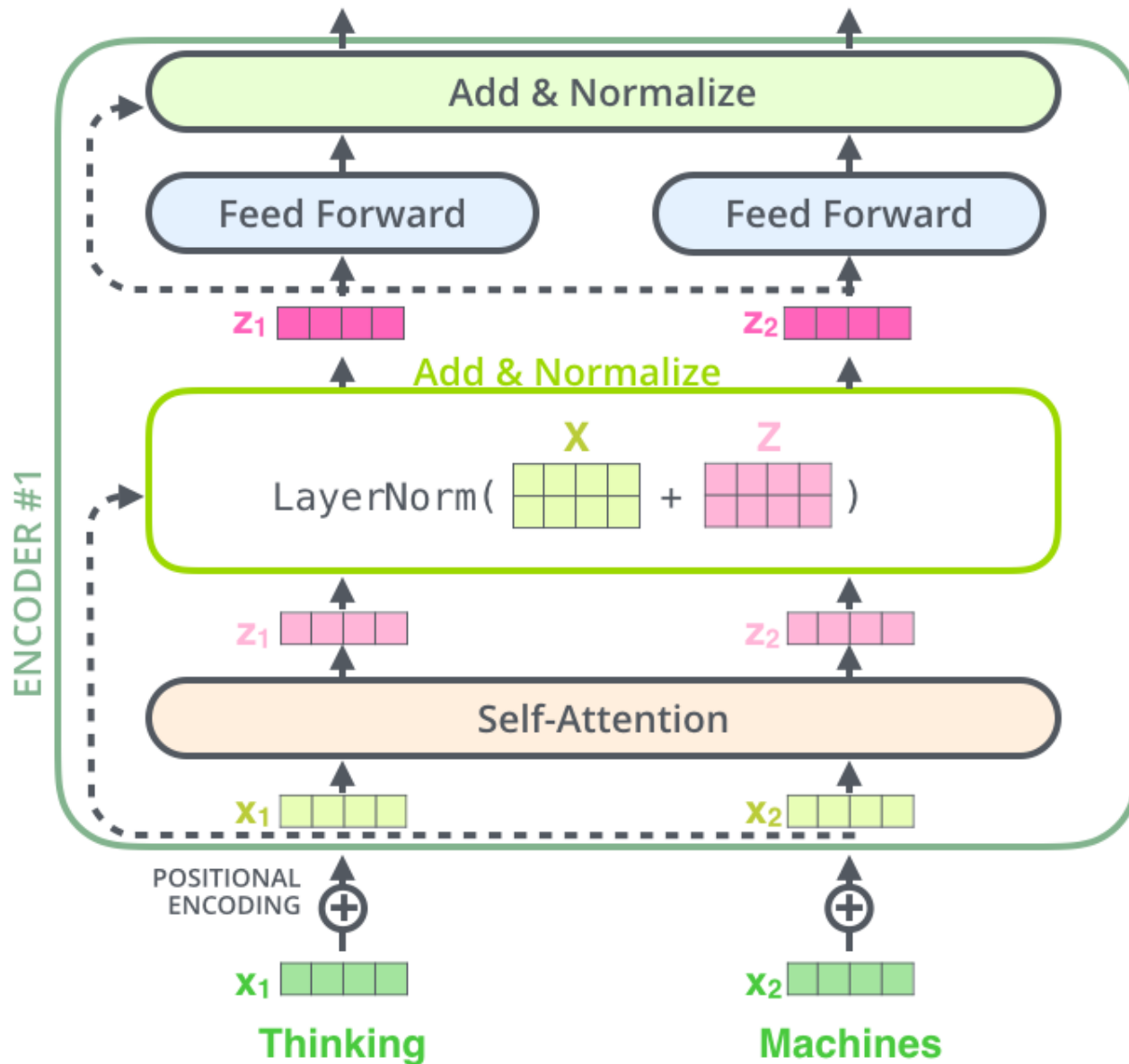


Linear

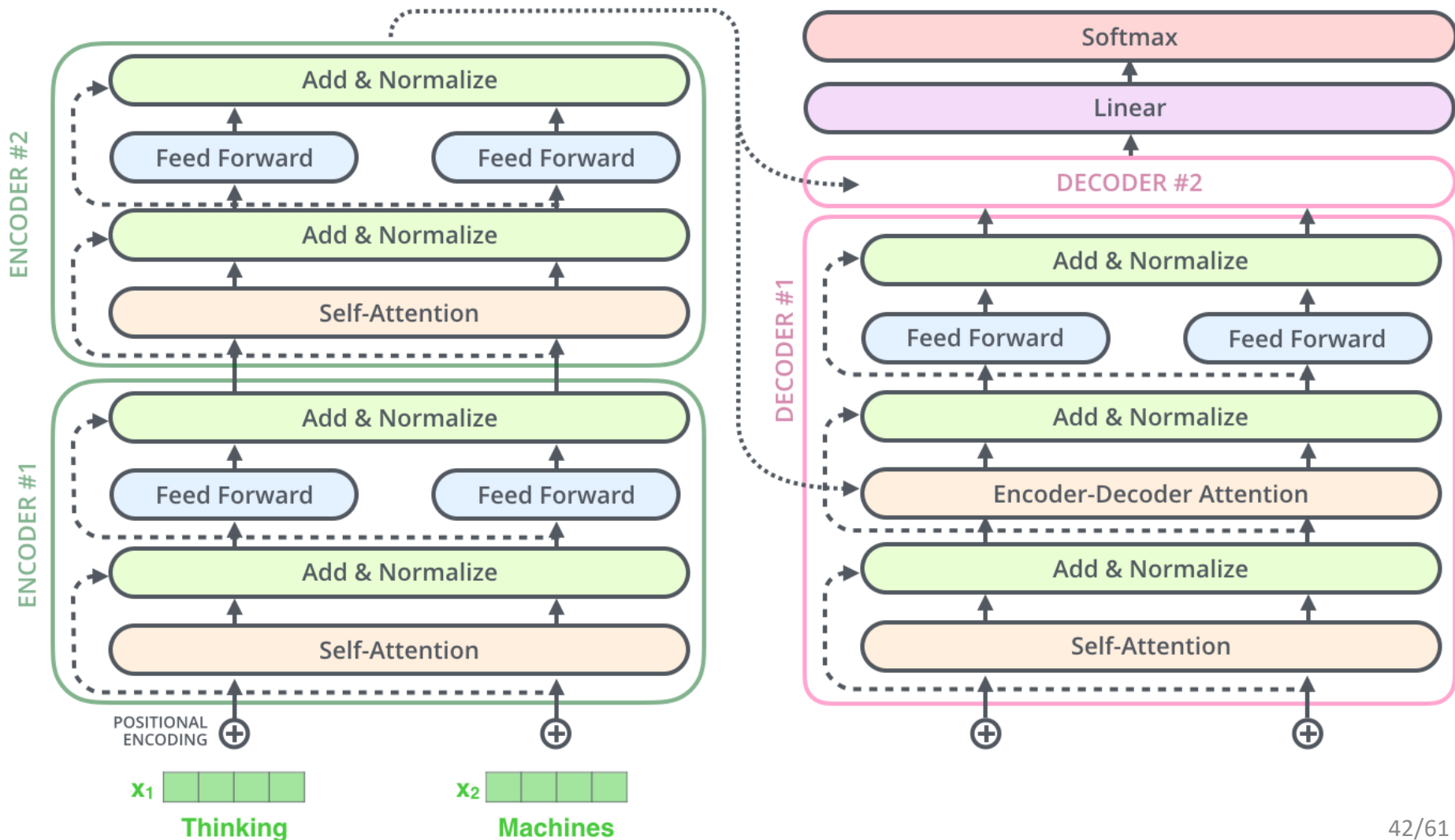
Decoder stack output



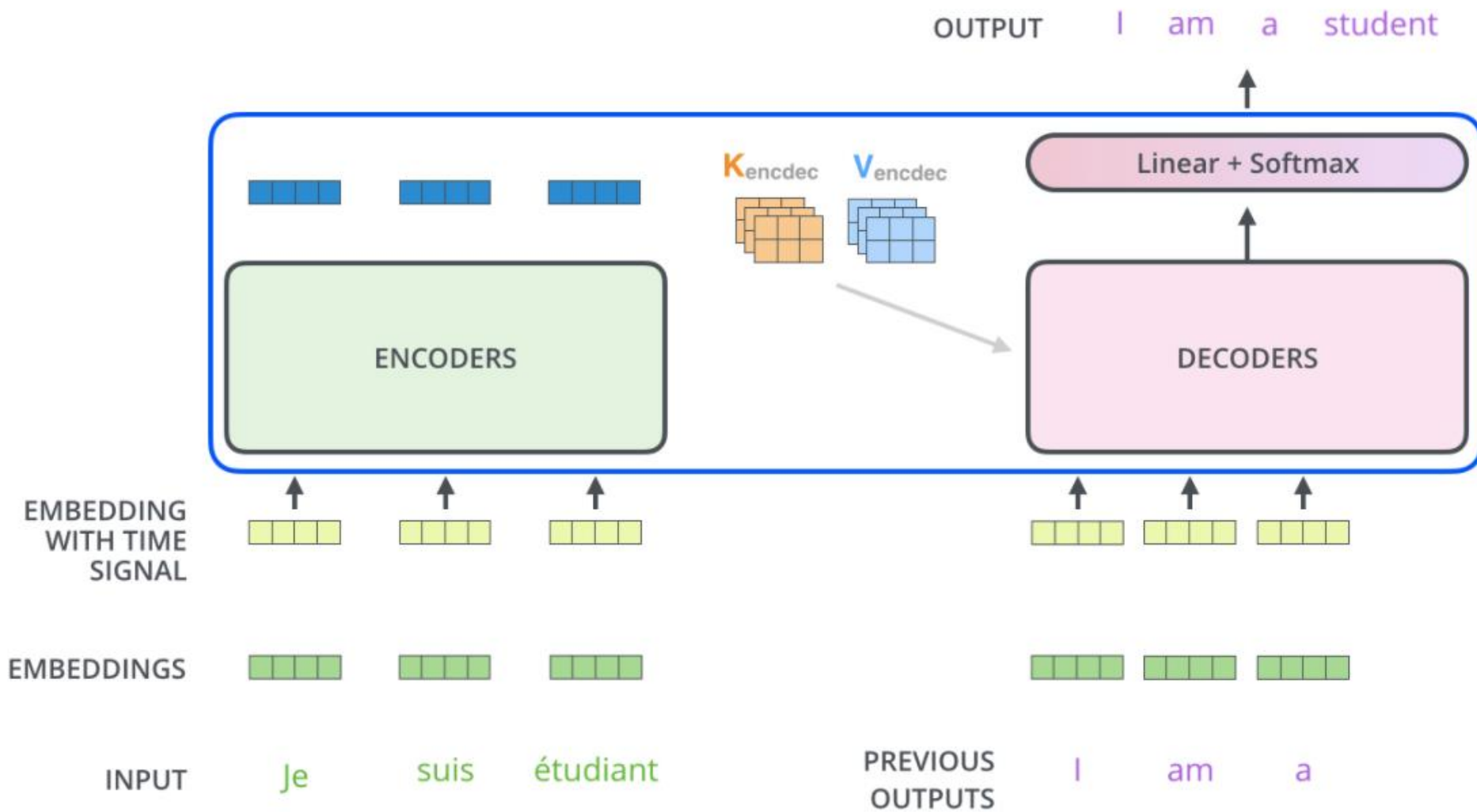
Transformer: Model Architecture



Transformer: Model Architecture



Transformer: Model Architecture



Transformer

Subword Tokenization: Byte Pair Encoding

- Проблема фиксированного словаря – невозможно представить несловарные слова (Out-Of-Vocabulary, OOV)
 - Токен <UNK>
- Byte Pair Encoding (BPE) – метод разбиения слов на токены:
 - частые слова представляются отдельными токенами
 - редкие и неизвестные слова разбиваются на несколько токенов (подслов)
 - Rico Sennrich, Barry Haddow, Alexandra Birch (University of Edinburgh). Neural Machine Translation of Rare Words with Subword Units // 54th Annual Meeting of the Association for Computational Linguistics (ACL). 2016
 - Philip Gage. A New Algorithm for Data Compression // The C User Journal. 1994

Subword Tokenization: Byte Pair Encoding

- Алгоритм BPE включает два этапа:
 - Обучение (training) – построение словаря токенов
 - Применение (inference) – использование словаря токенов для сегментации текста

ВРЕ: обучение

- На входе:
 - словарь токенов (token vocabulary) – содержит символы
 - задан максимальный размер словаря
 - таблица слияний (merge table) – пустая
- Алгоритм обучения:
 - Пока не достигли максимального размера словаря:
 - подсчитываем количество вхождений пар символов в корпус
 - находим наиболее частотную пару
 - сливаем пару – добавляем пару в таблицу и новый токен в словарь

BPE: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
c a t	4	a + t	19	All characters	
m a t	5	m + a	10		
m a t s	2	t + e	6		
m a t e	3	c + a	4		
a t e	3	t + s	2		
e a t	2	e + a	2		

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
c a t	4	a + t	19	All characters at	a t → at
m a t	5	m + a	10		
m a t s	2	t + e	6		
m a t e	3	c + a	4		
a t e	3	t + s	2		
e a t	2	e + a	2		

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
c at	4	m + at	10	All characters at	a t → at
m at	5	at + e	6		
m at s	2	c + at	4		
m at e	3	at + s	2		
at e	3	e + at	2		
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
c at	4	m + at	10	All characters at mat	a t → at
m at	5	at + e	6		m at → mat
m at s	2	c + at	4		
m at e	3	at + s	2		
at e	3	e + at	2		
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
c at	4	c + at	4	All characters at mat	a t $\rightarrow$ at
mat	5	at + e	3		m at $\rightarrow$ mat
mat s	2	mat + e	3		
mat e	3	mat + s	2		
at e	3	e + at	2		
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
c at	4	c + at	4	All characters at mat cat	a t → at
mat	5	at + e	3		m at → mat
mat s	2	mat + e	3		c at → cat
mat e	3	mat + s	2		
at e	3	e + at	2		
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
cat	4	at + e	3	All characters at mat cat	a t $\rightarrow$ at
mat	5	mat + e	3		m at $\rightarrow$ mat
mat s	2	mat + s	2		c at $\rightarrow$ cat
mat e	3	e + at	2		
at e	3				
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
cat	4	at + e	3	All characters at mat cat ate	a t → at
mat	5	mat + e	3		m at → mat
mat s	2	mat + s	2		c at → cat
mat e	3	e + at	2		at e → ate
at e	3				
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
cat	4	mat + e	3	All characters at mat cat ate mate	a t → at
mat	5	mat + s	2		m at → mat
mat s	2	e + at	2		c at → cat
mat e	3				at e → ate
ate	3				m ate → mate
e at	2				

ВРЕ: обучение

Corpus		Token pairs		Token vocabulary	Merge table
words	count	pair	count		
cat	4	mat + s	2	All characters at mat cat ate mate	a t → at
mat	5	e + at	2		m at → mat
mat s	2				c at → cat
mate	3				at e → ate
ate	3				m ate → mate
e at	2				

Остановка – достигли максимального размера словаря

ВРЕ: применение

Corpus: tokenization		Token vocabulary	Merge table
words	tokens		
cat	cat	All characters at mat cat ate mate	a t → at
mat	mat		m at → mat
mats	mat ##s		c at → cat
mate	mate		at e → ate
ate	ate		m ate → mate
eat	e ##at		

s и **##s** – разные токены

Byte-level BPE

- В качестве исходных токенов рассматриваются байты (256), а не символы
- Можно не использовать токен <UNK>
- Используется в GPT-2

WordPiece

- Максимизируется не частота совместной встречаемости токенов (как в BPE), а правдоподобие:

$$score = \frac{f(ab)}{f(a)f(b)},$$

где $f(ab)$ – частота пары токенов ab ,

$f(a), f(b)$ – частота токенов a и b

- Используется в BERT
- Schuster M., Nakajima K. Japanese and Korean voice search // 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (2012)

Transformer

- Lena Voita – NLP Course: Transformer
 - https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html#transformer_intro
- Jay Alammam – The Illustrated Transformer
 - <http://jalammar.github.io/illustrated-transformer/>
- HarvardNLP – The Annotated Transformer
 - <http://nlp.seas.harvard.edu/annotated-transformer/>
- Dive into Deep Learning – Attention Mechanisms
 - https://d2l.ai/chapter_attention-mechanisms-and-transformers/
- Александр Дьяконов – Токенизация на подслова
 - <https://dyakonov.org/2019/11/29/токенизация-на-подслова-subword-tokenization/>
- Hugging Face – Summary of the tokenizers
 - https://huggingface.co/docs/transformers/tokenizer_summary

BERT

– Bidirectional Encoder
Representations
from Transformers



Оригинальная статья

arXiv > cs > arXiv:1810.04805

[Submitted on 11 Oct 2018 (v1), last revised 24 May 2019 (this version, v2)]

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova

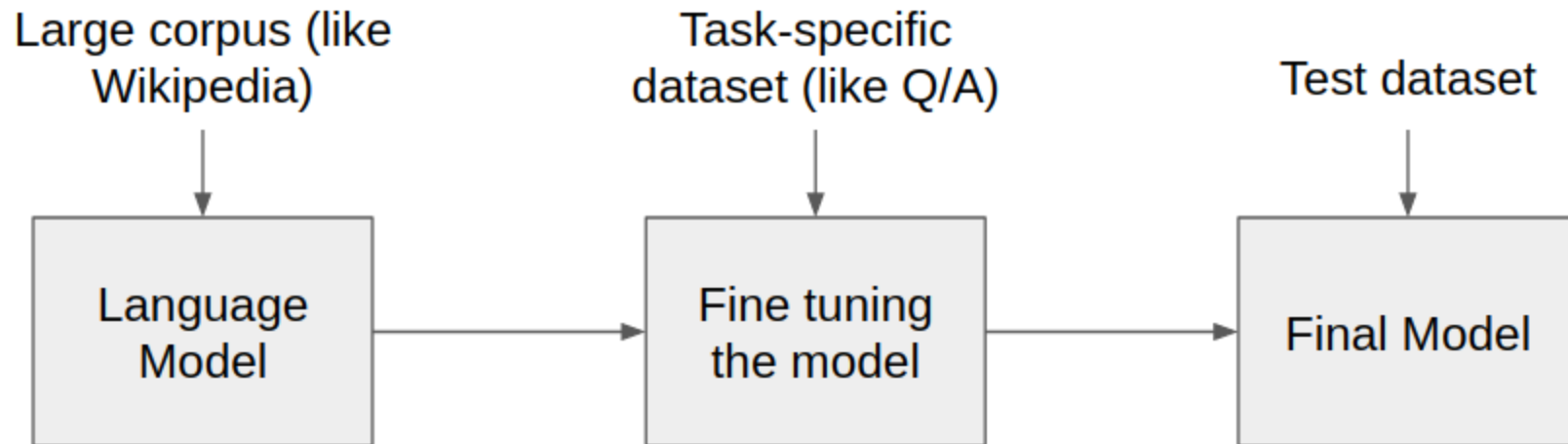
We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

Боле 82 000 ссылок в Google Scholar

Полезные ссылки

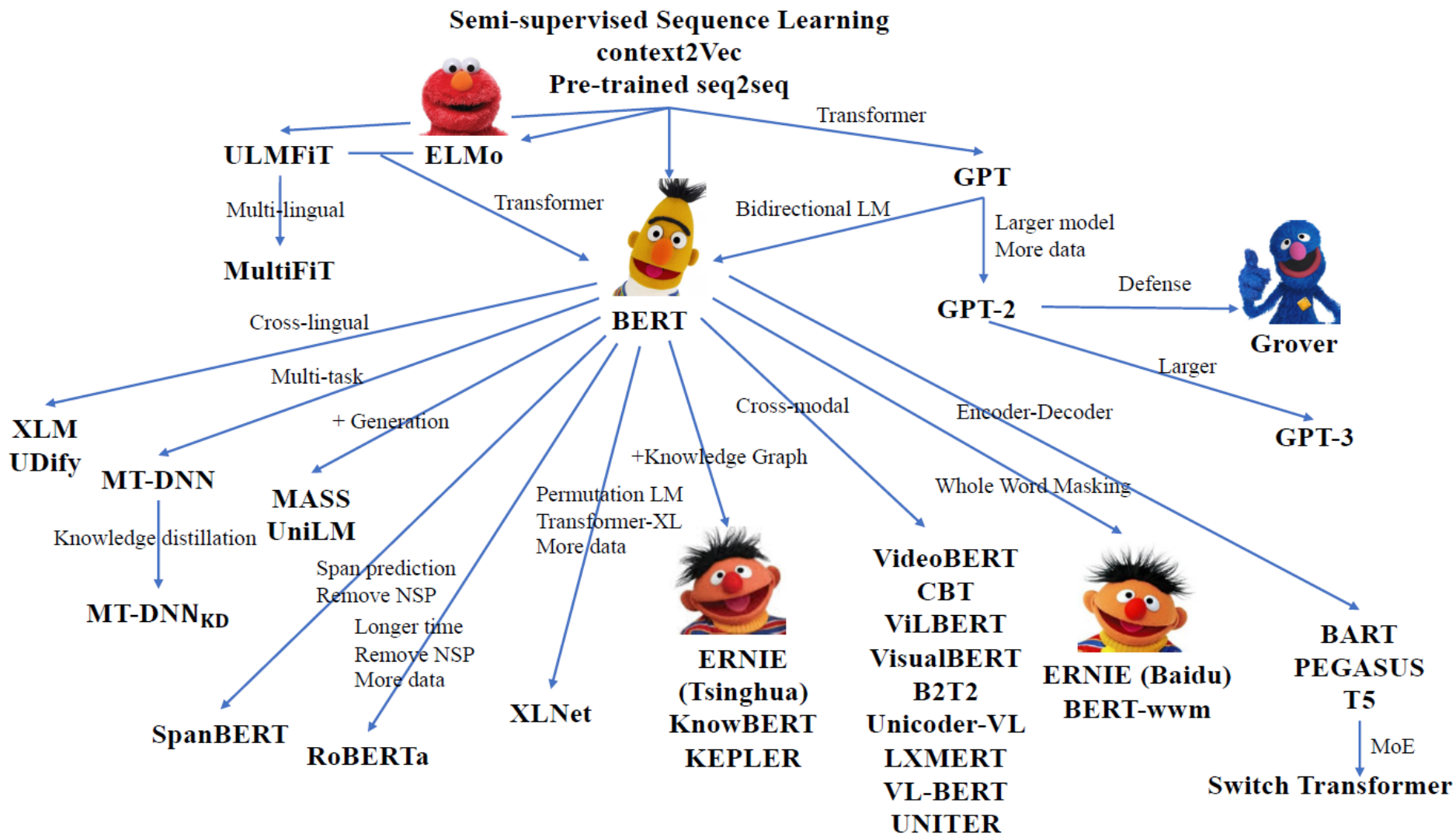
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (Google AI Language) – BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // NAACL, 2019
 - <https://aclanthology.org/N19-1423>
- The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
 - Оригинал: <http://jalamar.github.io/illustrated-bert>
 - Перевод: <https://habr.com/ru/post/487358>
- Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework
 - <https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework>
- Репозиторий Google Research на GitHub
 - <https://github.com/google-research/bert>
- Тьюриал по запуску BERT локально и на Google Colab
 - <https://habr.com/ru/post/436878>

Языковые модели



Transfer Learning in NLP = Pre-Training and Fine-Tuning

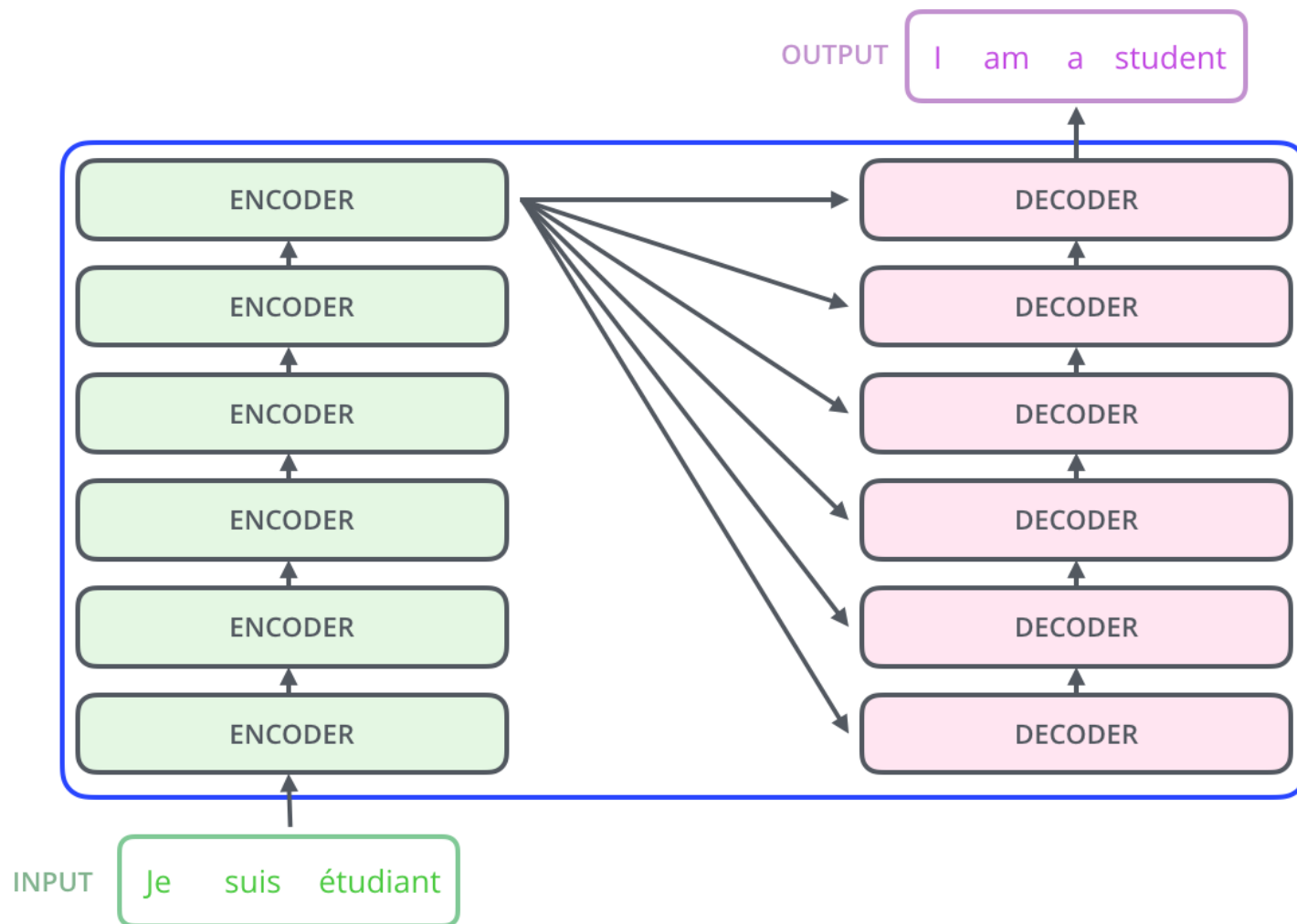
Языковые модели



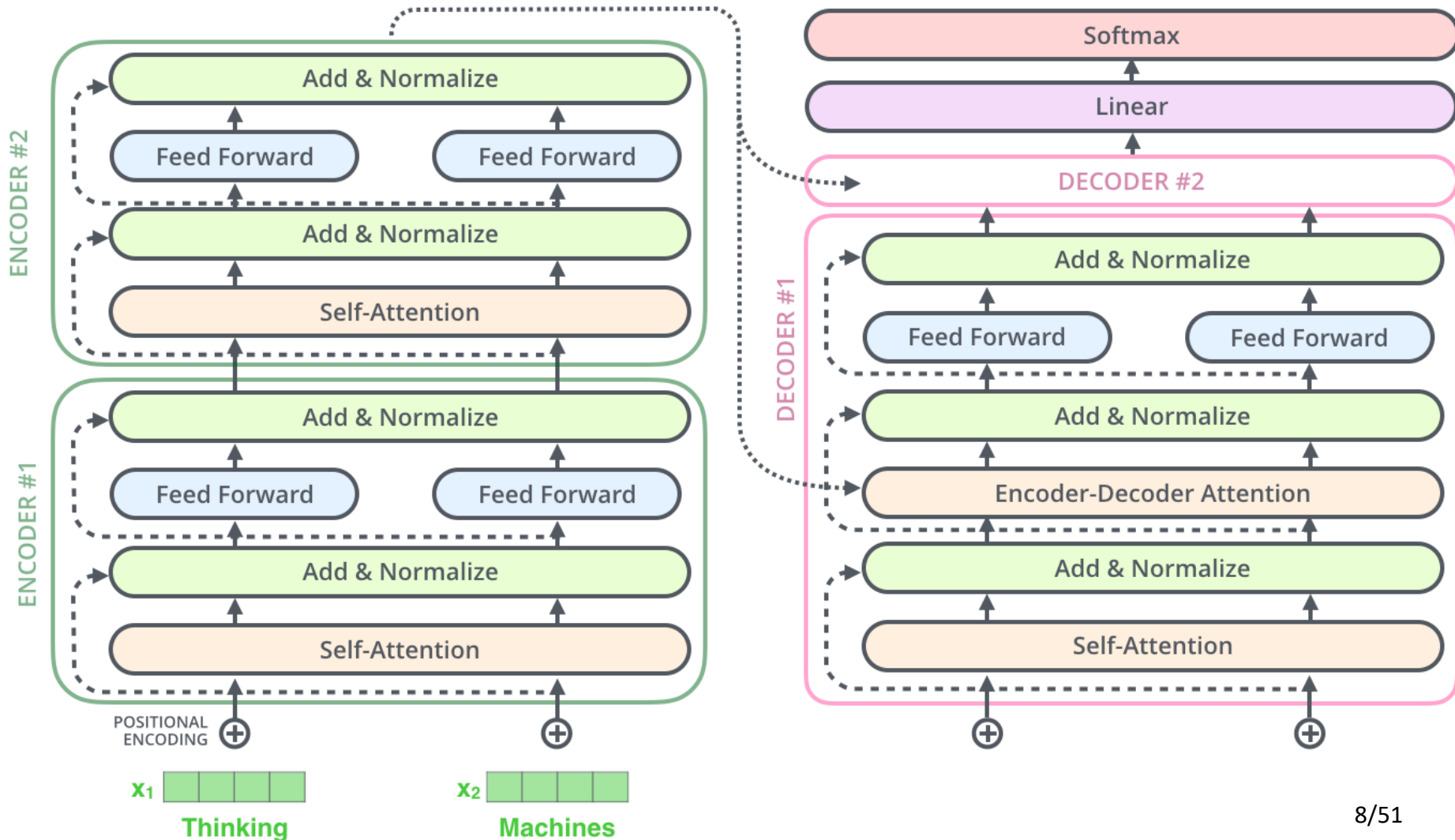
Transformer

- Vaswani et al. (Google Brain) – Attention Is All You Need // NIPS, 2017
 - <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- The Illustrated Transformer
 - Оригинал: <https://jalammar.github.io/illustrated-transformer/>
 - Перевод: <https://habr.com/ru/post/486358/>
- Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)
 - Оригинал: <https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/>
 - Перевод: <https://habr.com/ru/post/486158/>

Transformer: архитектура



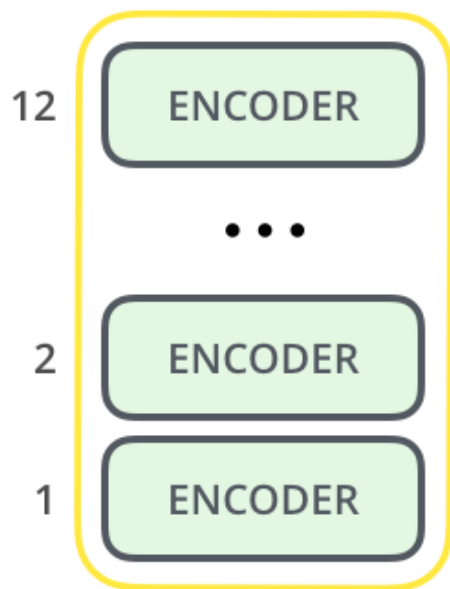
Transformer: архитектура



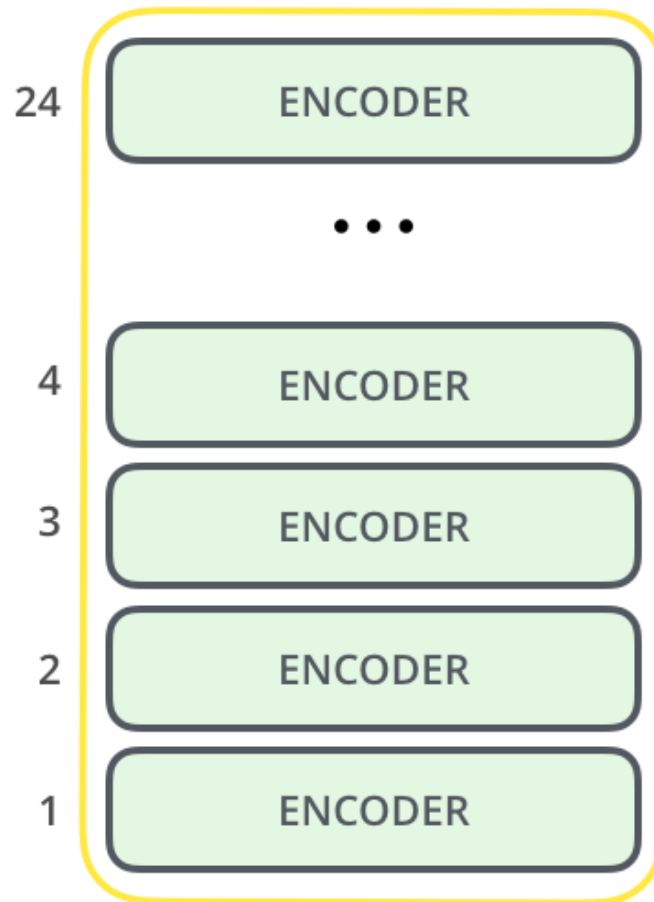
BERT: архитектура

340 млн параметров

110 млн параметров

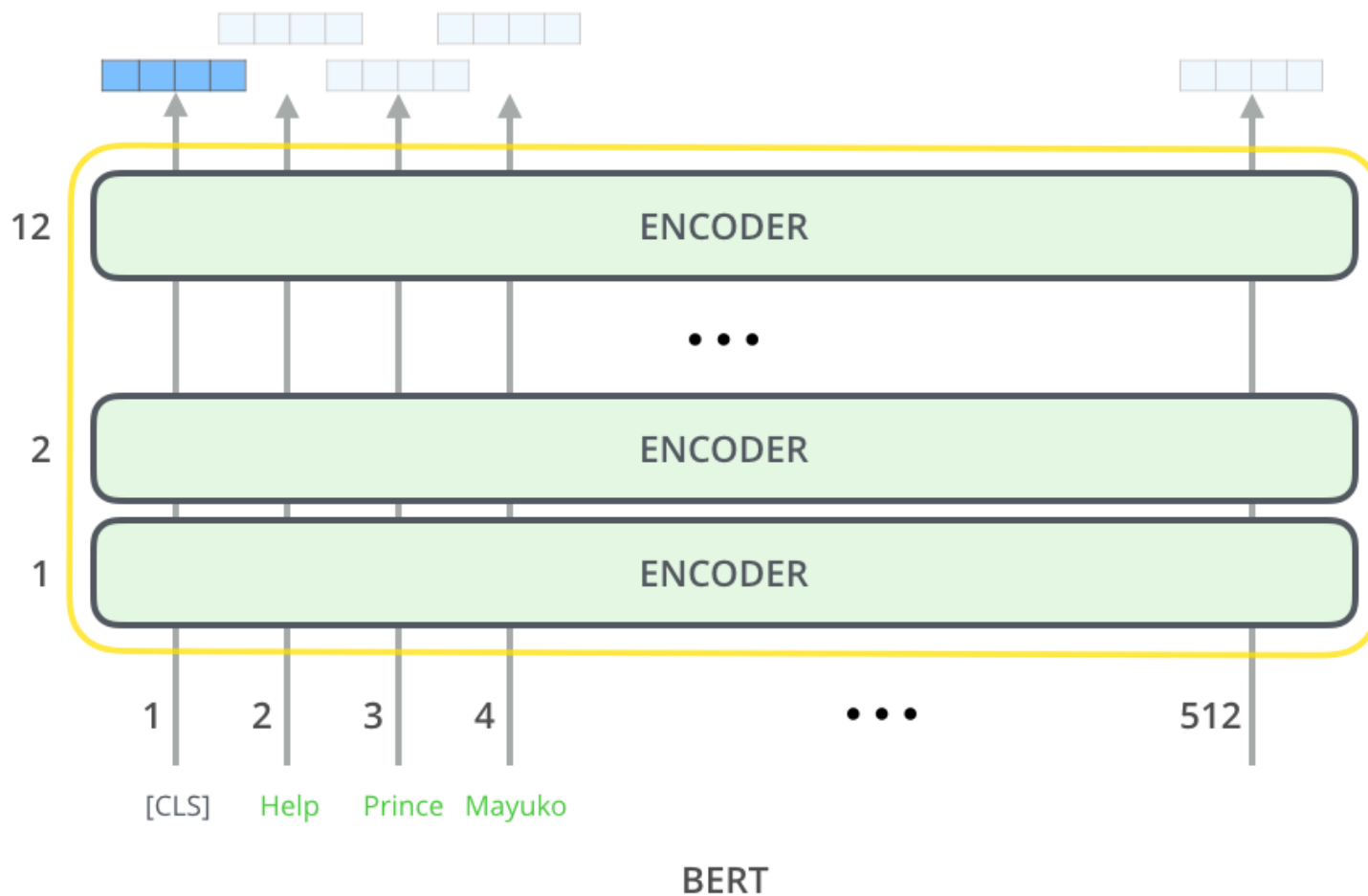


BERT_{BASE}

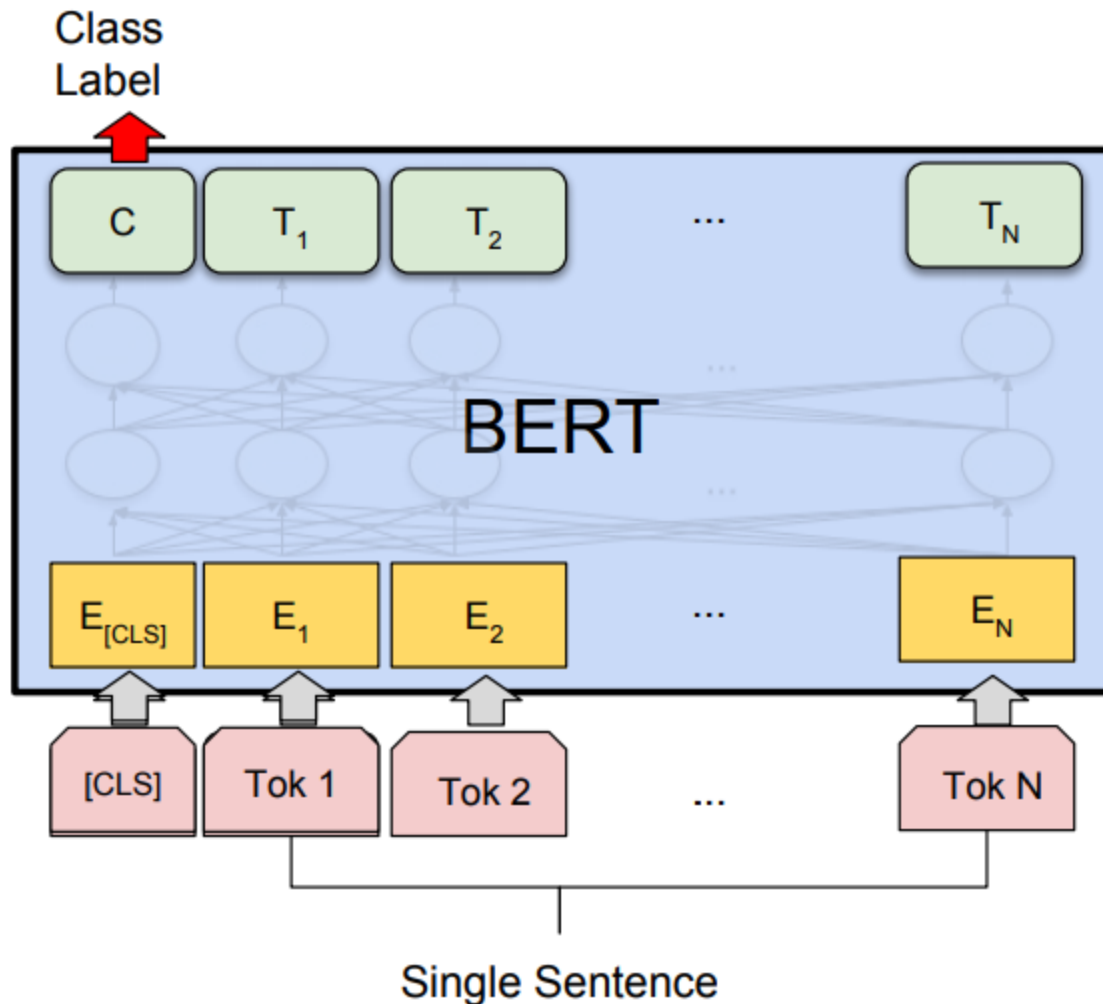


BERT_{LARGE}

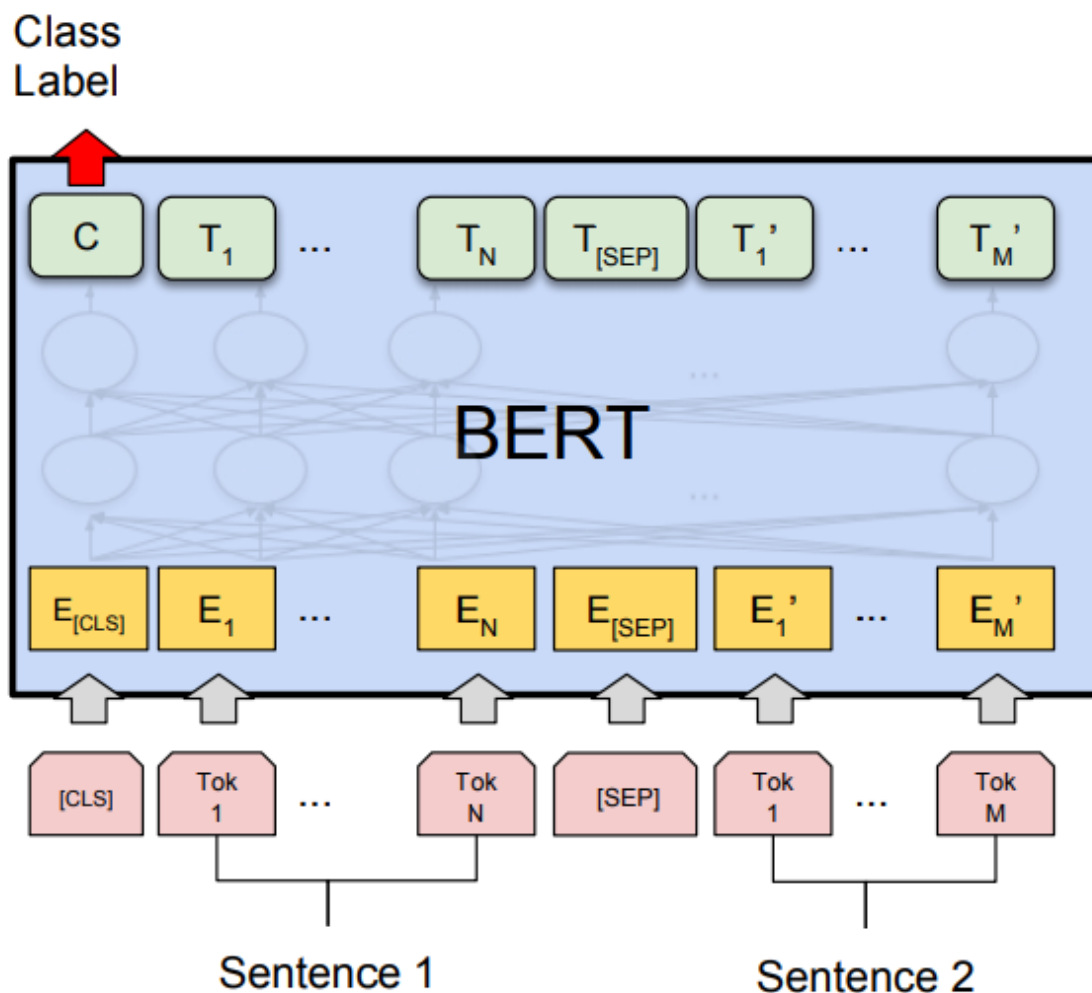
BERT: архитектура



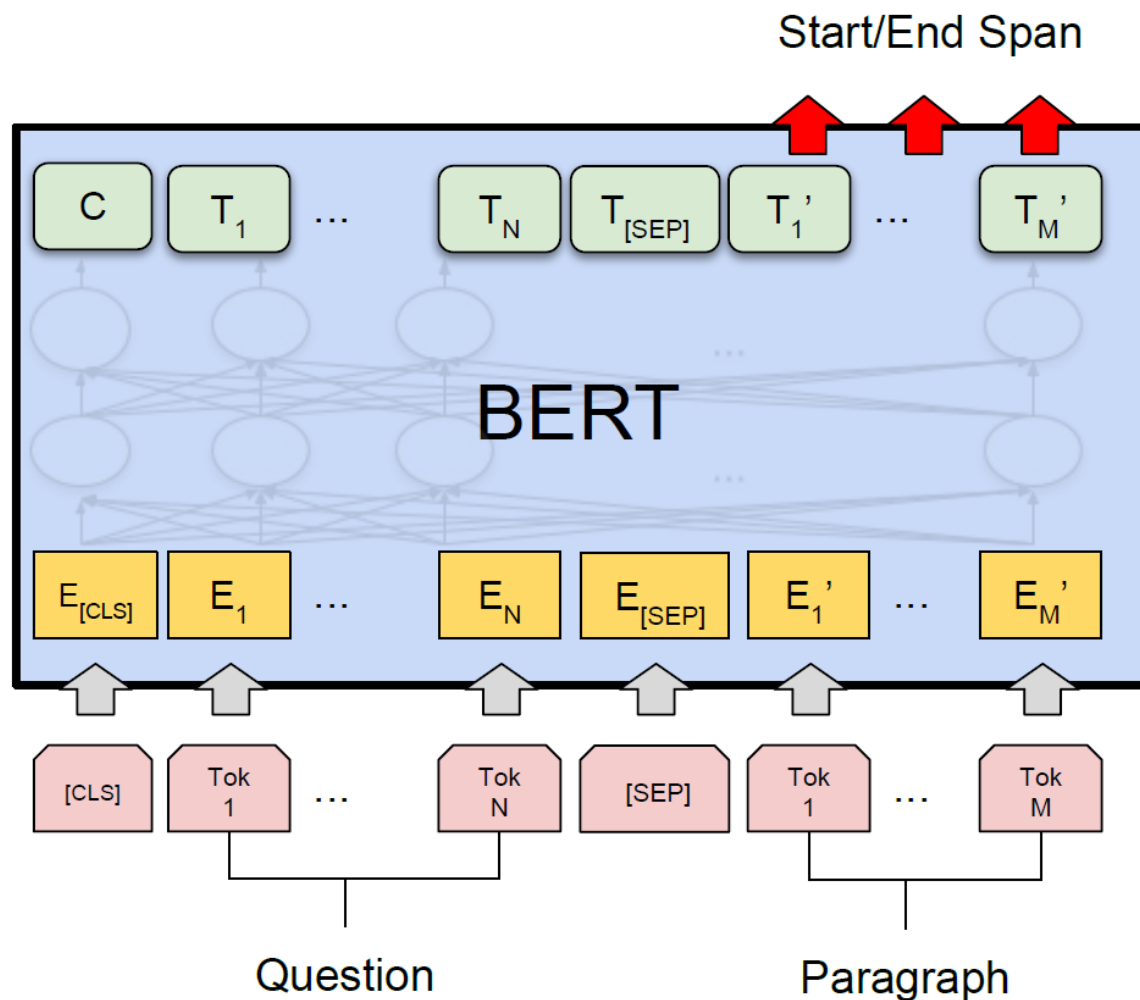
BERT: классификация одного предложения



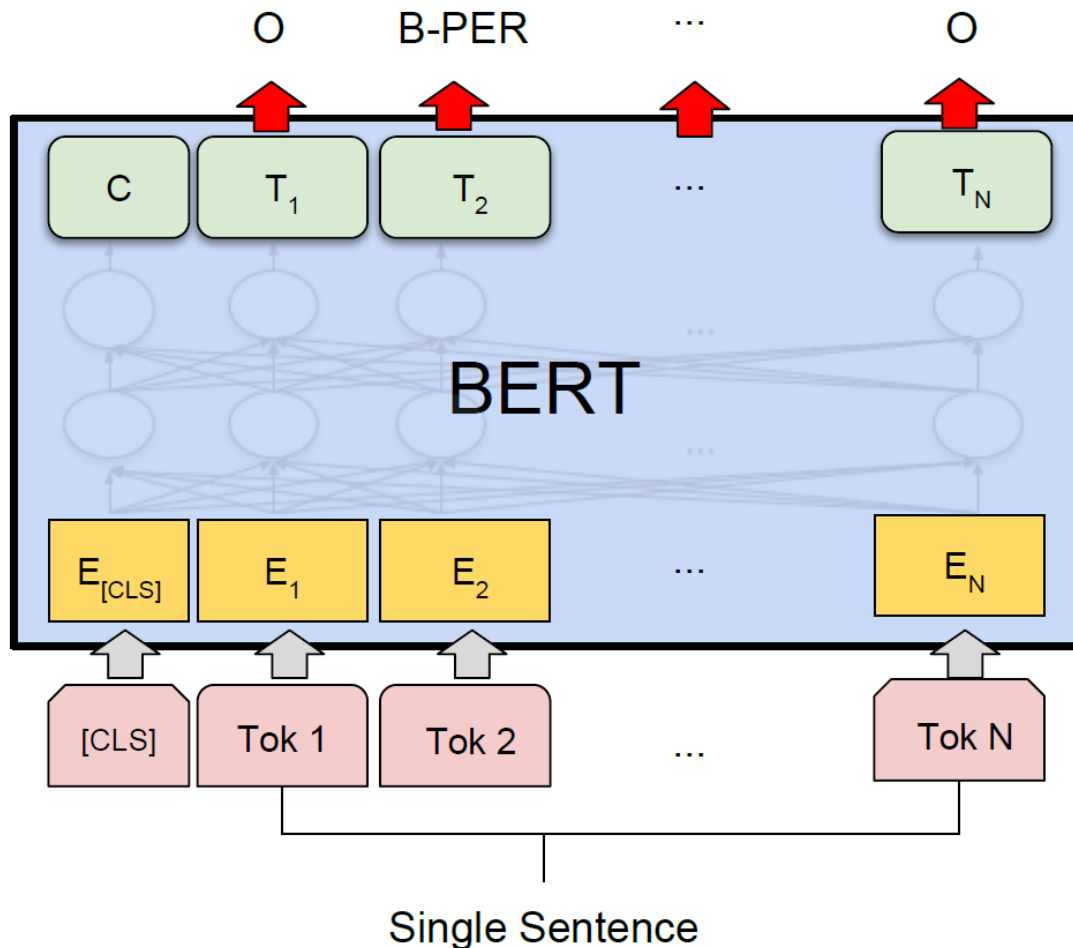
BERT – классификация пары предложений



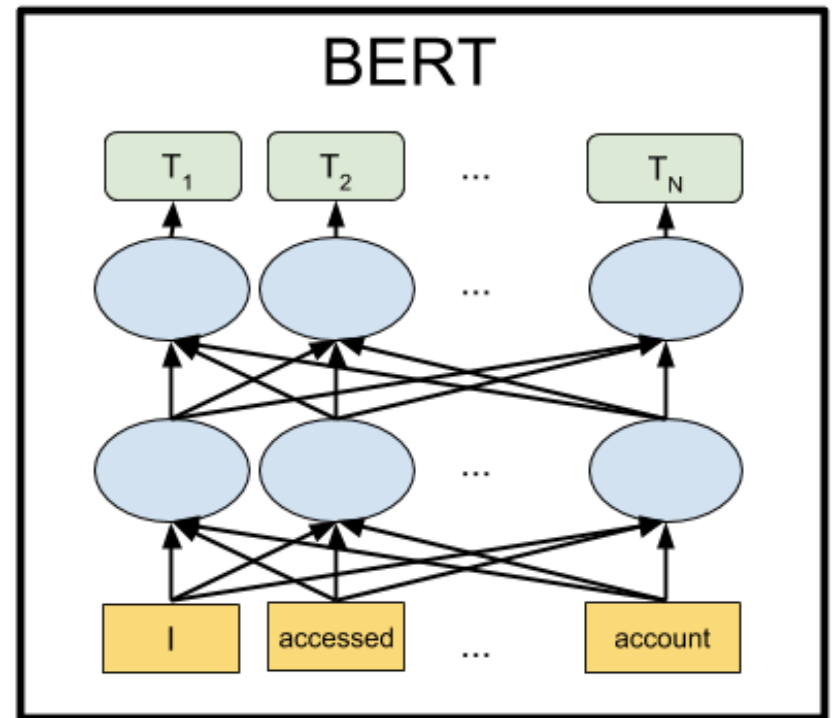
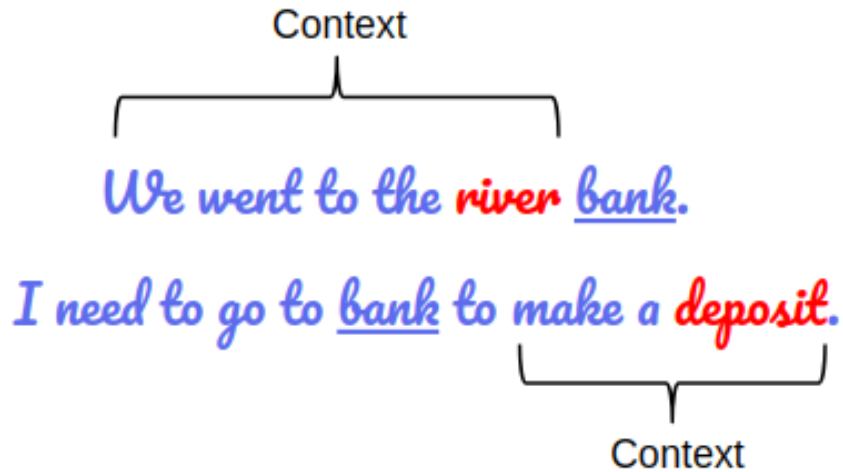
BERT – вопросно-ответная модель (Q/A)



BERT – извлечение именованных сущностей (Named Entity Recognition, NER)



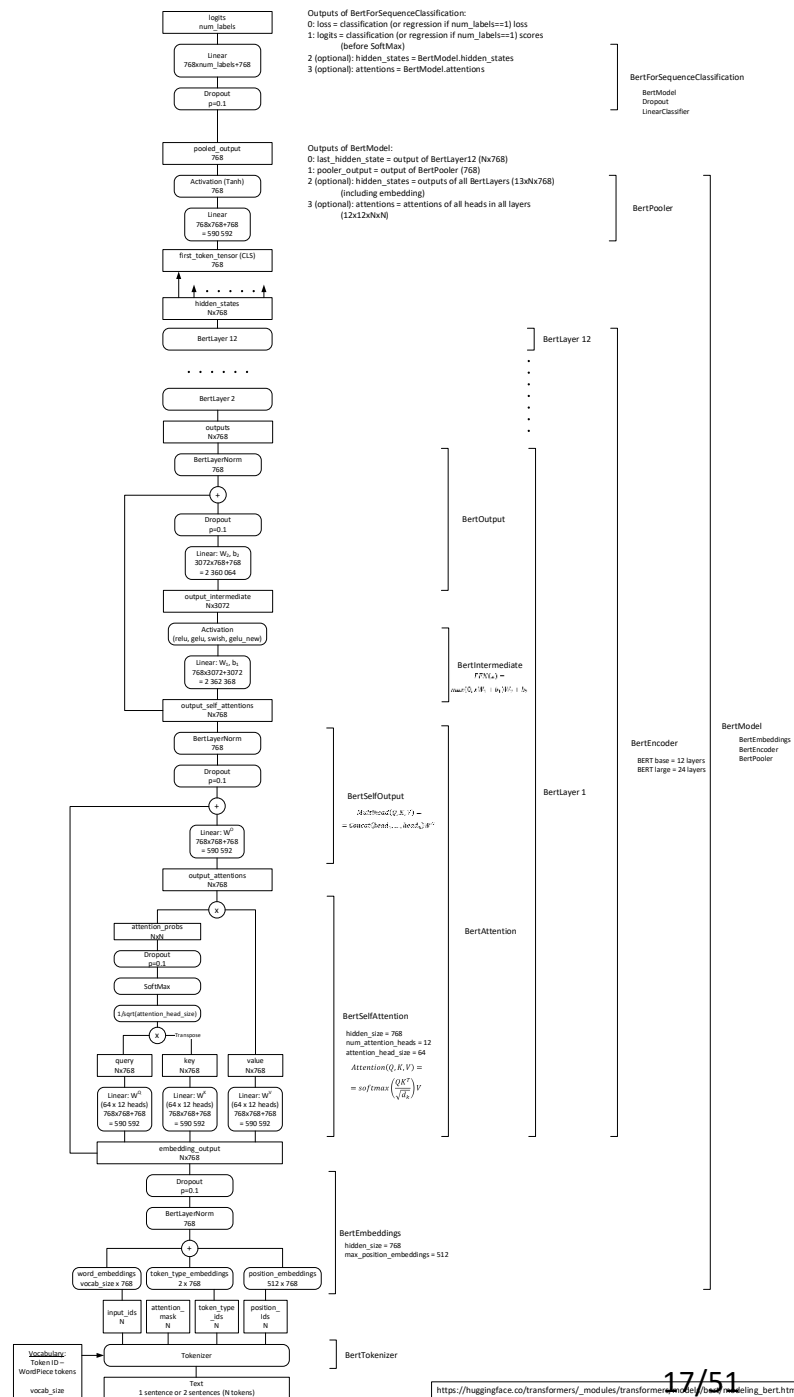
BERT: Bidirectional Encoder



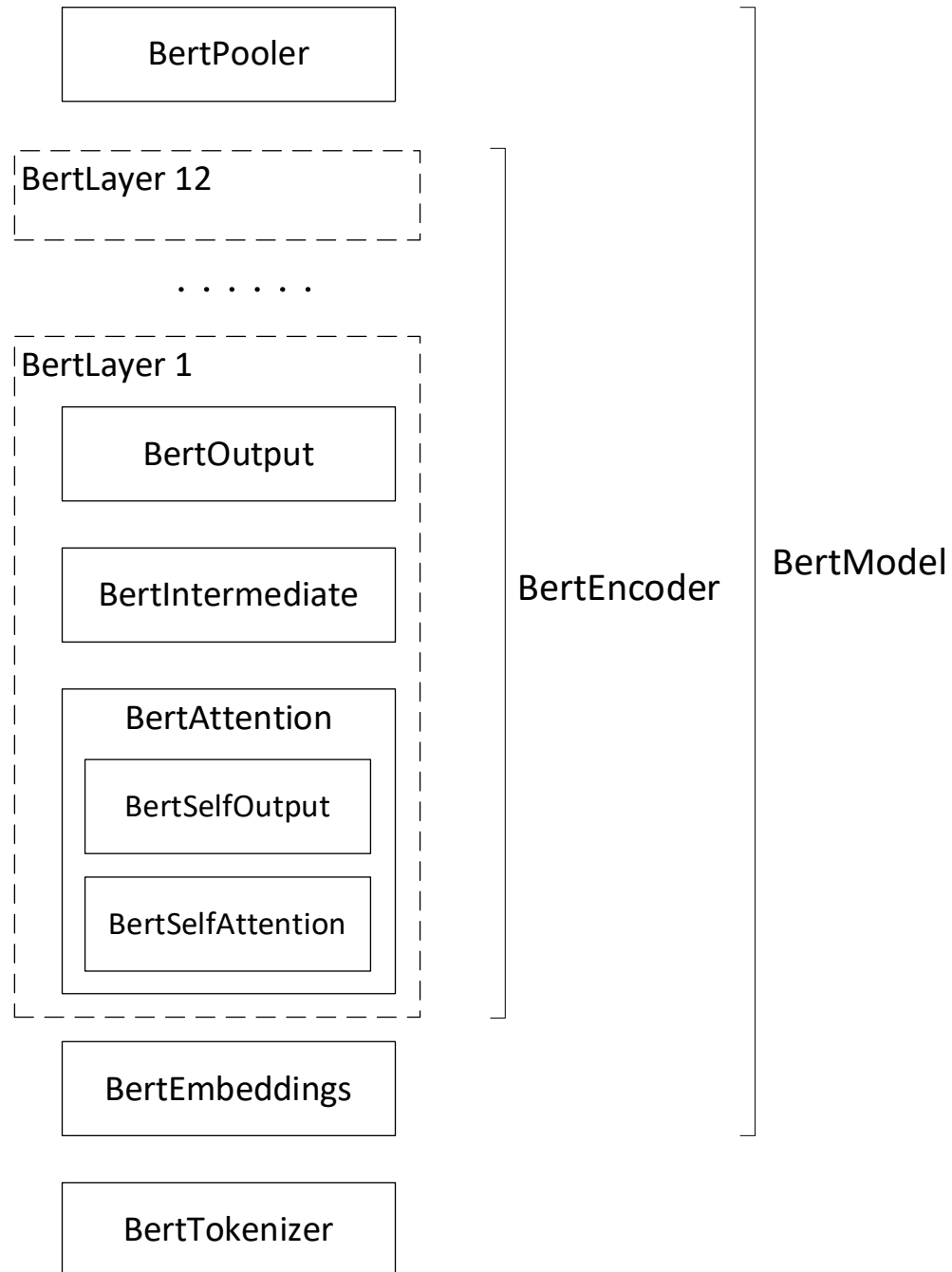
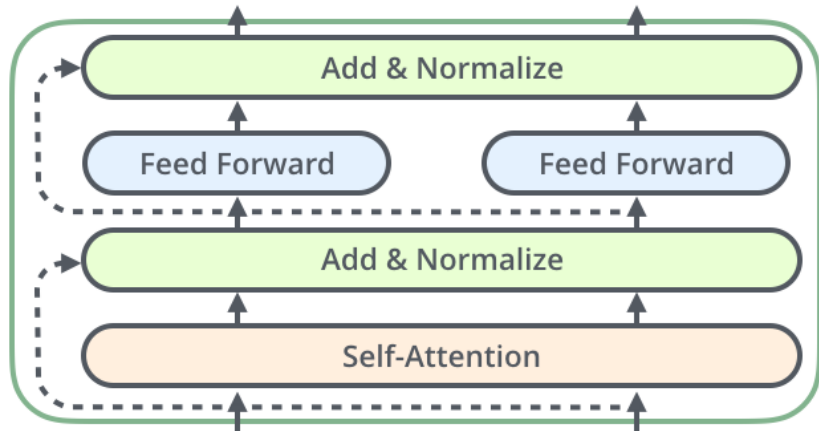
BERT: задачи для обучения

- Masked Language Modeling (MLM)
 - Предсказание 15% случайно выбранных замаскированных слов
- Next Sentence Prediction (NSP)
 - Предсказание следующего предложения
- Оригинальный BERT обучен на следующих корпусах:
 - BooksCorpus (800M words)
 - English Wikipedia (2,500M words)

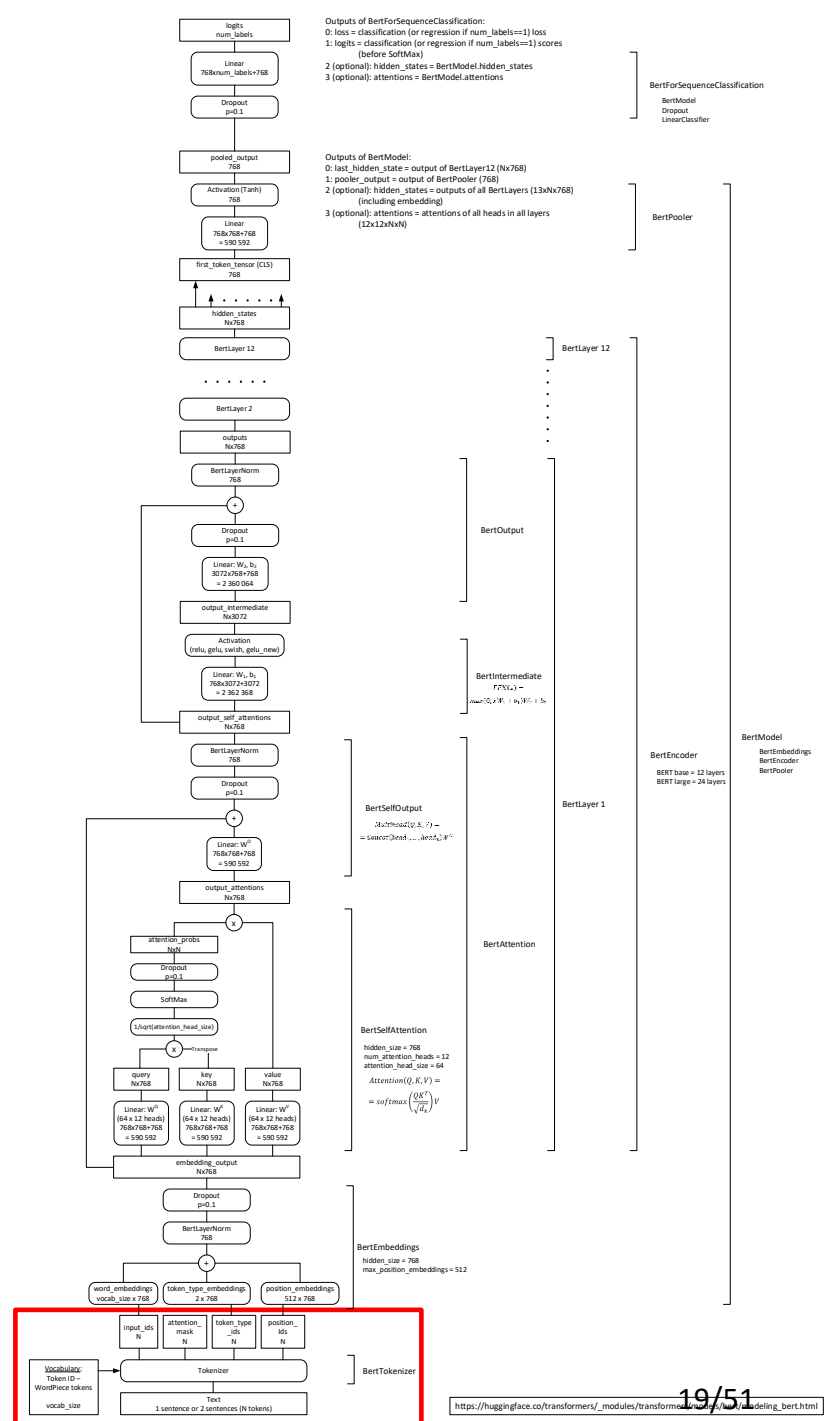
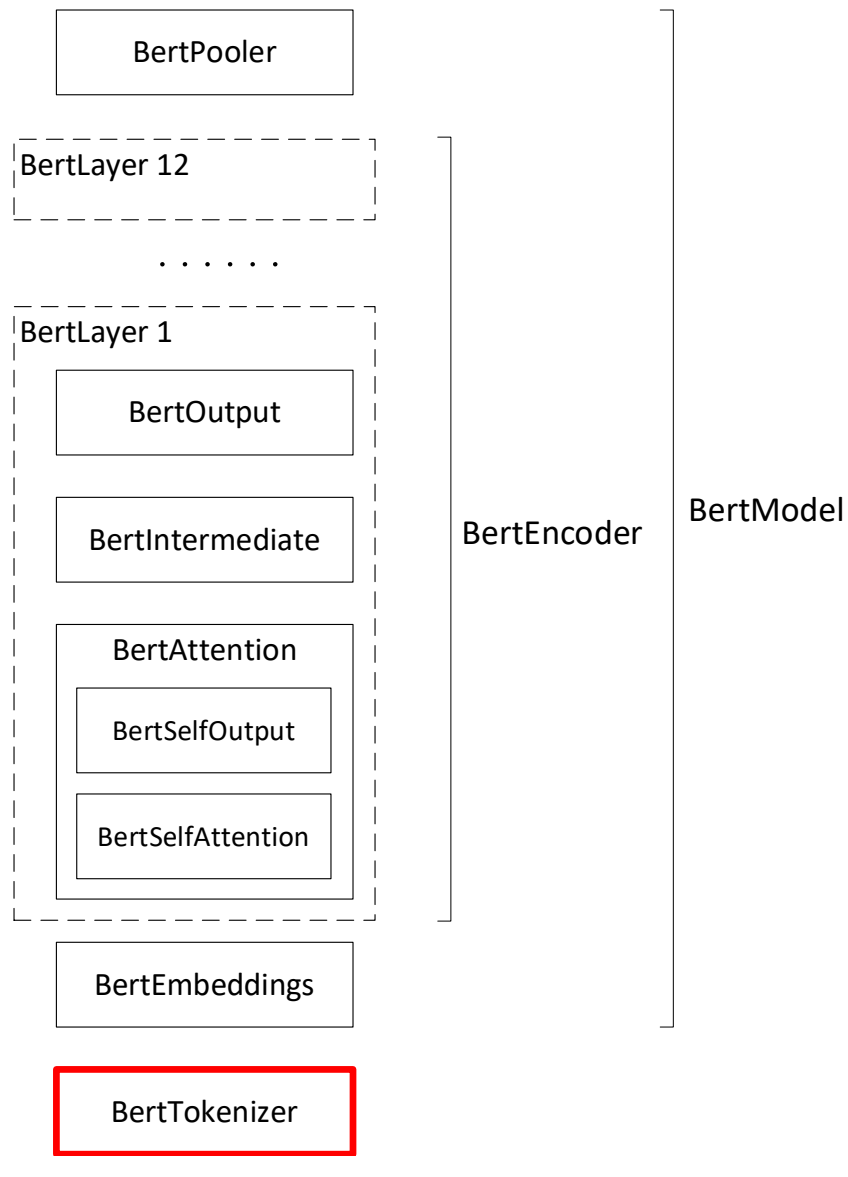
Архитектура BERT



Архитектура BERT



Токенизация



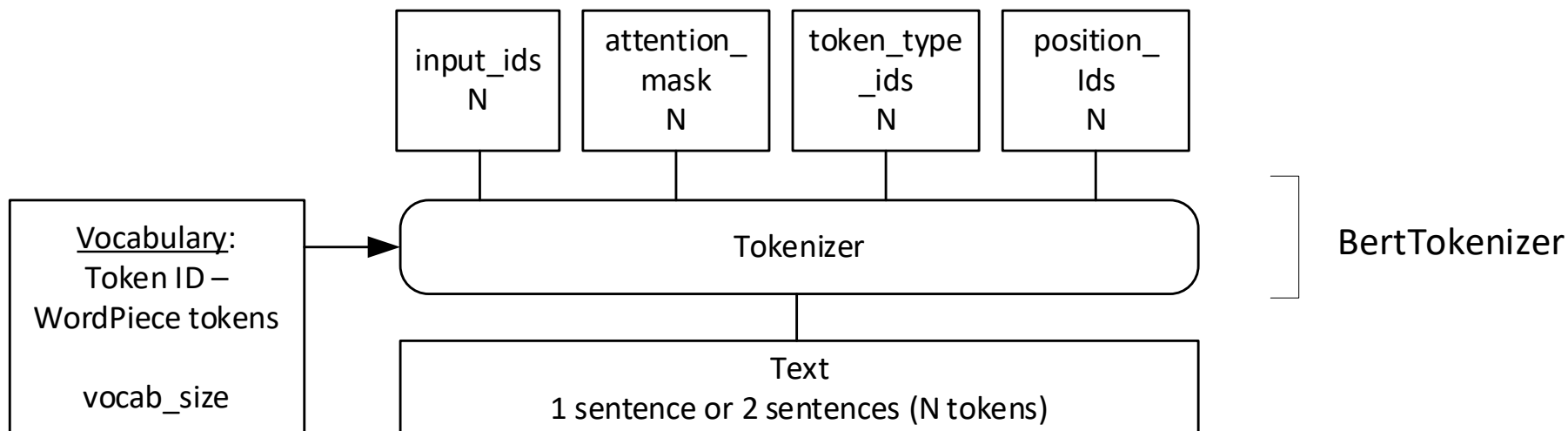
Токенизация: WordPiece

- WordPiece tokens
 - Schuster M., Nakajima K. Japanese and Korean voice search // 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (2012)
 - Wu et al. – Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation // arXiv:1609.08144 (2016)
- «чудесный день!» → [“чудесный”, “день”, “!”]
- «Чудесный день!» → [“Чудес”, “##ный”, “день”, “!”]
 - cased model – регистрозависимая модель
 - uncased model – регистронеависимая модель

Токенизация: WordPiece

- Отличия от BPE:
 - Алгоритм:
 - BPE: максимизируется частота совместной встречаемости токенов
 - WordPiece: максимизируется правдоподобие
 - Разбиение:
 - BPE: walking → walk@@ + ing
 - WordPiece: walking → walk + ##ing

Токенизация



Токенизация

- `input_ids` – список ID токенов (из словаря)
- `attention_mask` – бинарная маска для отделения токенов от заполнителя (PAD)
- `token_type_ids` – бинарная маска для отделения первого предложения (включая начальный токен CLS и разделитель SEP) (0) от второго (включая конечный SEP) (1)
- `position_ids` – номера позиций токенов

Токенизация

Sentence A

Я устал

Sentence B

Пойду посплю

Токенизация

Sentence A

Я устал

Sentence B

Пойду посплю

WordPiece токены

Я

устал

Пойду

пос

##плю

Токенизация

Sentence A

Я устал

Sentence B

Пойду посплю

WordPiece токены

Я

устал

Пойду

пос

##плю

Специальные
токены

[CLS]

Я

устал

[SEP]

Пойду

пос

##плю

[SEP]

[PAD]

Токенизация

	Sentence A			Sentence B				
	Я устал			Пойду посплю				
WordPiece токены	Я	устал		Пойду	пос	##плю		
Специальные токены	[CLS]	Я	устал	[SEP]	Пойду	пос	##плю	[SEP] [PAD]
input_ids	101	478	19 896	102	26 525	2 430	14 063	102 0

Токенизация

	Sentence A			Sentence B				
	Я устал			Пойду посплю				
WordPiece токены	Я	устал		Пойду	пос	##плю		
Специальные токены	[CLS]	Я	устал	[SEP]	Пойду	пос	##плю	[SEP] [PAD]
input_ids	101	478	19 896	102	26 525	2 430	14 063	102 0
attention_mask	1	1	1	1	1	1	1	0

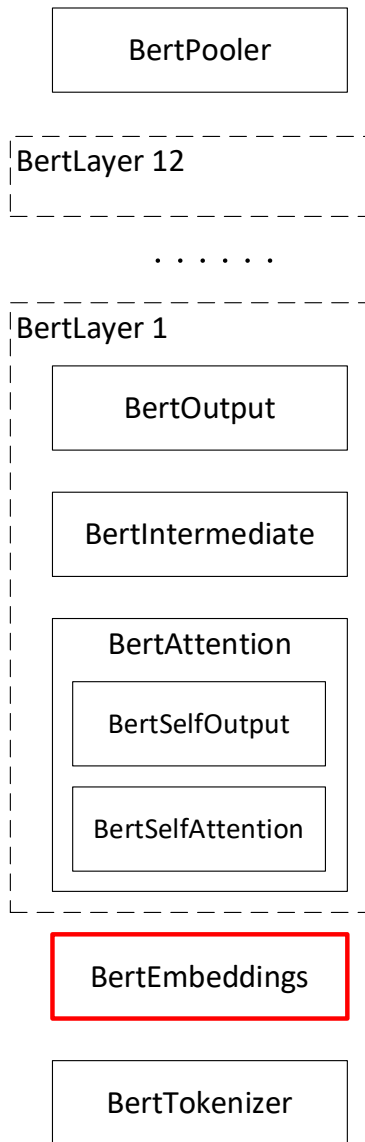
Токенизация

	Sentence A				Sentence B			
	Я устал				Пойду посплю			
WordPiece токены	Я		устал		Пойду		пос ##плю	
Специальные токены	[CLS]	Я	устал	[SEP]	Пойду	пос	##плю	[SEP] [PAD]
input_ids	101	478	19 896	102	26 525	2 430	14 063	102 0
attention_mask	1	1	1	1	1	1	1	1 0
token_type_ids	0	0	0	0	1	1	1	1 0

Токенизация

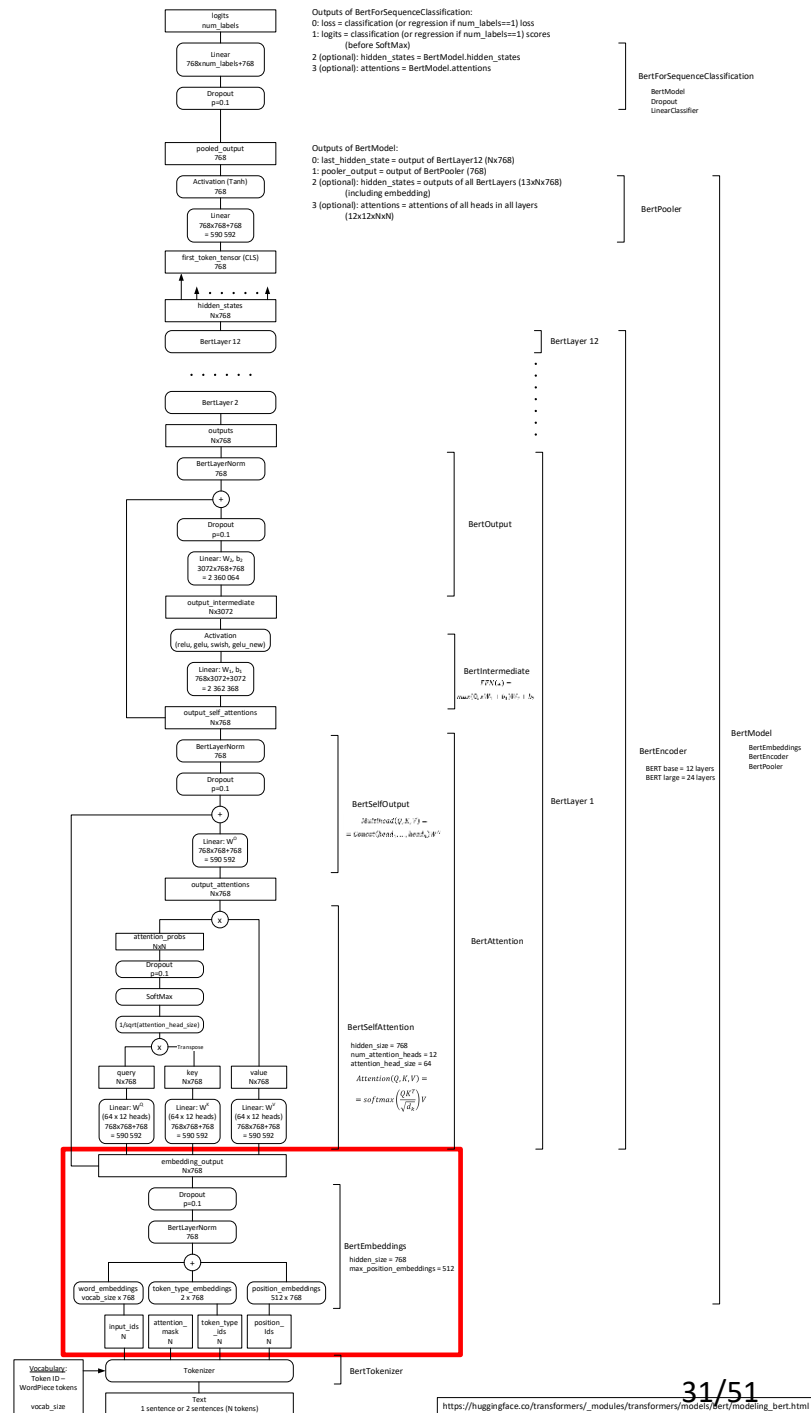
	Sentence A				Sentence B			
	Я устал				Пойду посплю			
WordPiece токены	Я		устал		Пойду		пос ##плю	
Специальные токены	[CLS]	Я	устал	[SEP]	Пойду	пос	##плю	[SEP] [PAD]
input_ids	101	478	19 896	102	26 525	2 430	14 063	102 0
attention_mask	1	1	1	1	1	1	1	1 0
token_type_ids	0	0	0	0	1	1	1	1 0
position_ids	0	1	2	3	4	5	6	7 8

Embeddings

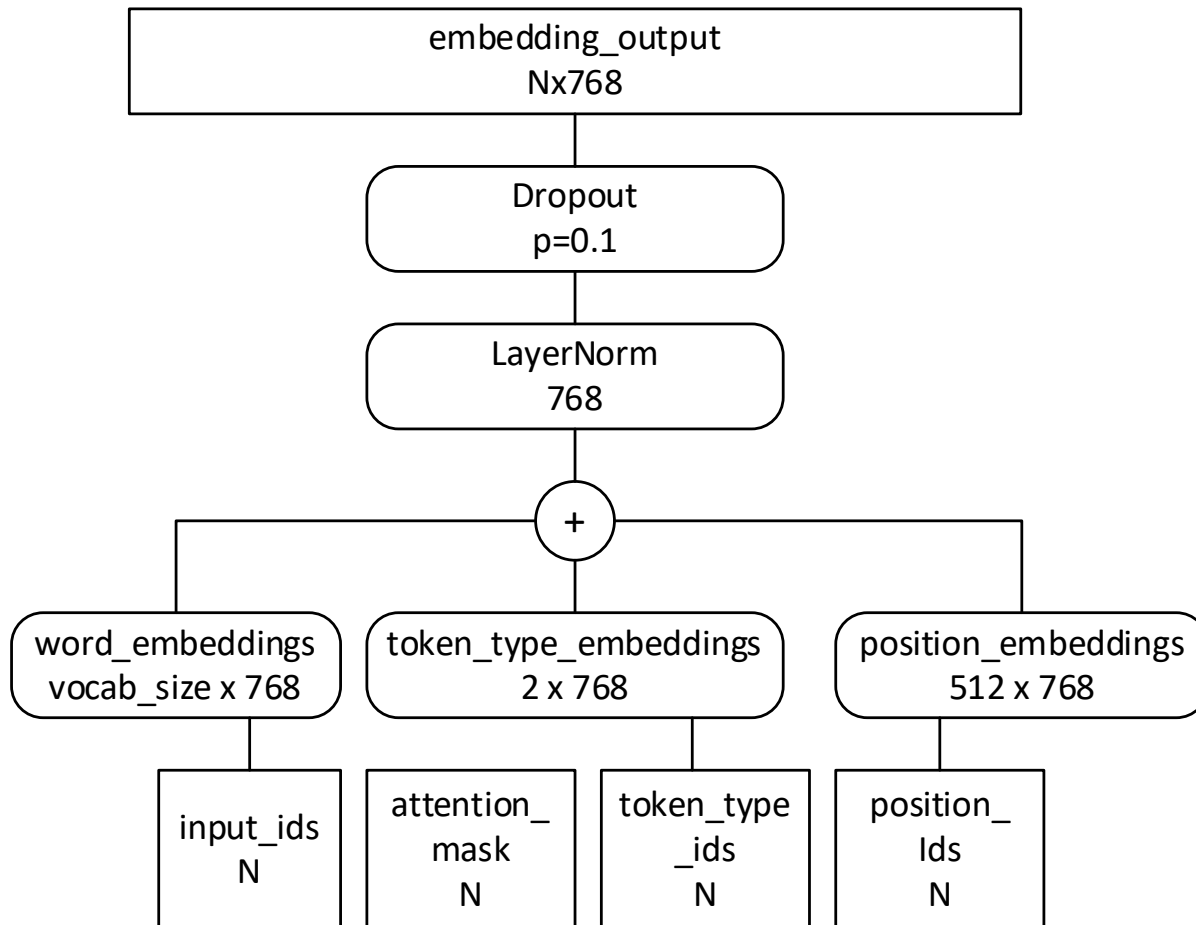


BertEncoder

BertModel



Embeddings

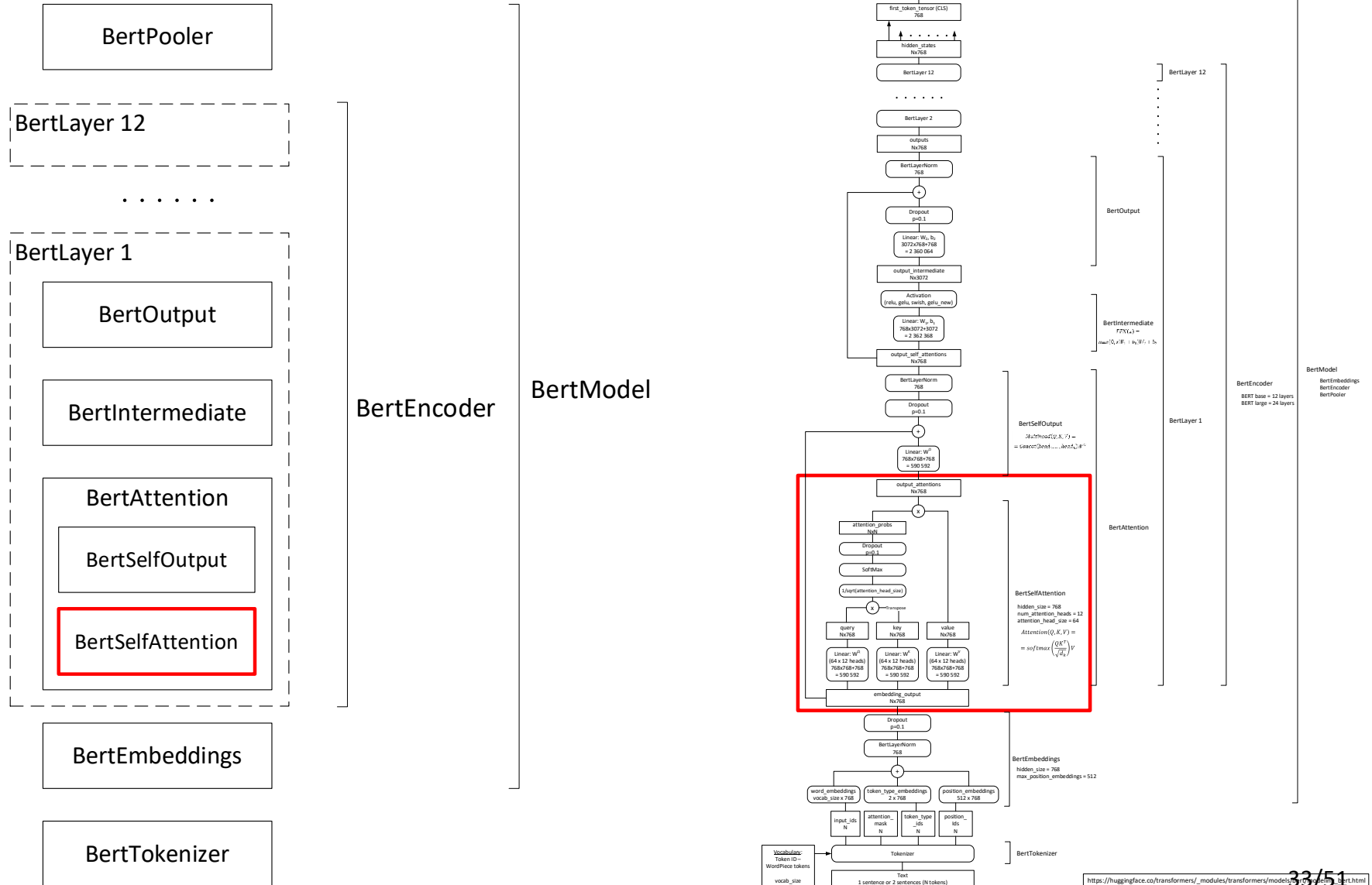


BertEmbeddings

hidden_size = 768

max_position_embeddings = 512

Self-Attention



Self-Attention

- Внимание
 - Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate (2014)
- Визуализируя нейронный машинный перевод (seq2seq модели с механизмом внимания)
 - <https://habr.com/ru/post/486158/>
- How do Transformers Work in NLP?
 - <https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/>
- The Illustrated Transformer
 - <http://jalammar.github.io/illustrated-transformer/>
- Getting meaning from text: self-attention step-by-step video
 - <https://peltarion.com/blog/data-science/self-attention-video>

Self-Attention

Each vector receives three representations (“roles”)

$$\begin{bmatrix} W_Q \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{blue} \\ \text{blue} \\ \text{blue} \end{bmatrix}$$

Query: vector **from** which the attention is looking

“Hey there, do you have this information?”

$$\begin{bmatrix} W_K \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{yellow} \\ \text{yellow} \\ \text{yellow} \end{bmatrix}$$

Key: vector **at** which the query looks to compute weights

“Hi, I have this information – give me a large weight!”

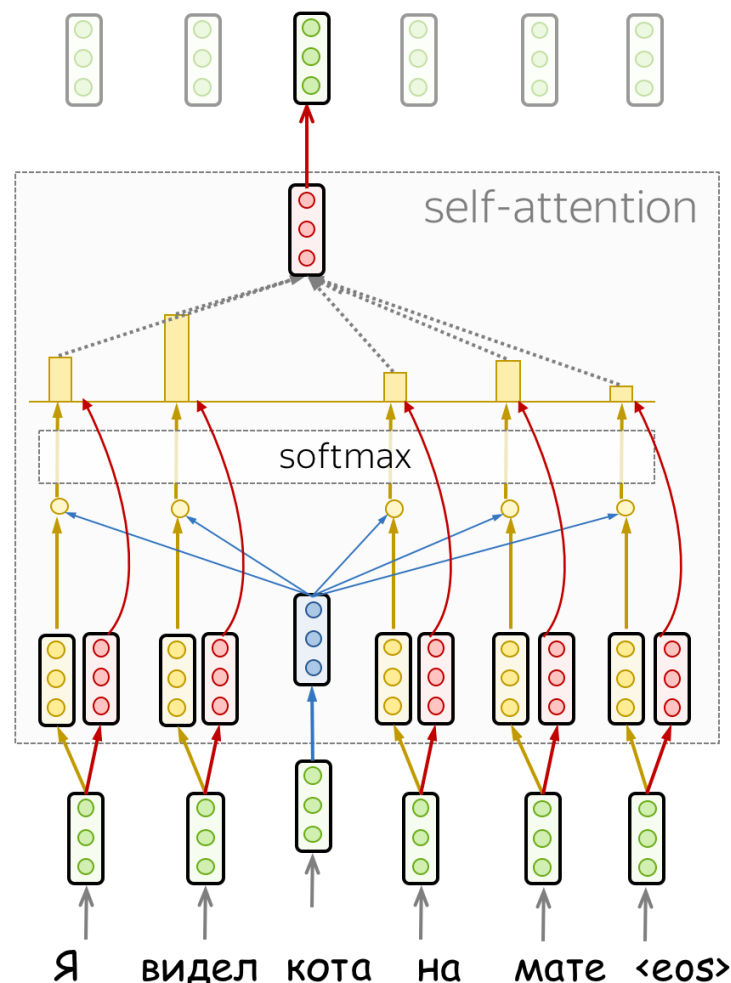
$$\begin{bmatrix} W_V \end{bmatrix} \times \begin{bmatrix} \text{green} \\ \text{green} \\ \text{green} \end{bmatrix} = \begin{bmatrix} \text{red} \\ \text{red} \\ \text{red} \end{bmatrix}$$

Value: their weighted sum is attention output

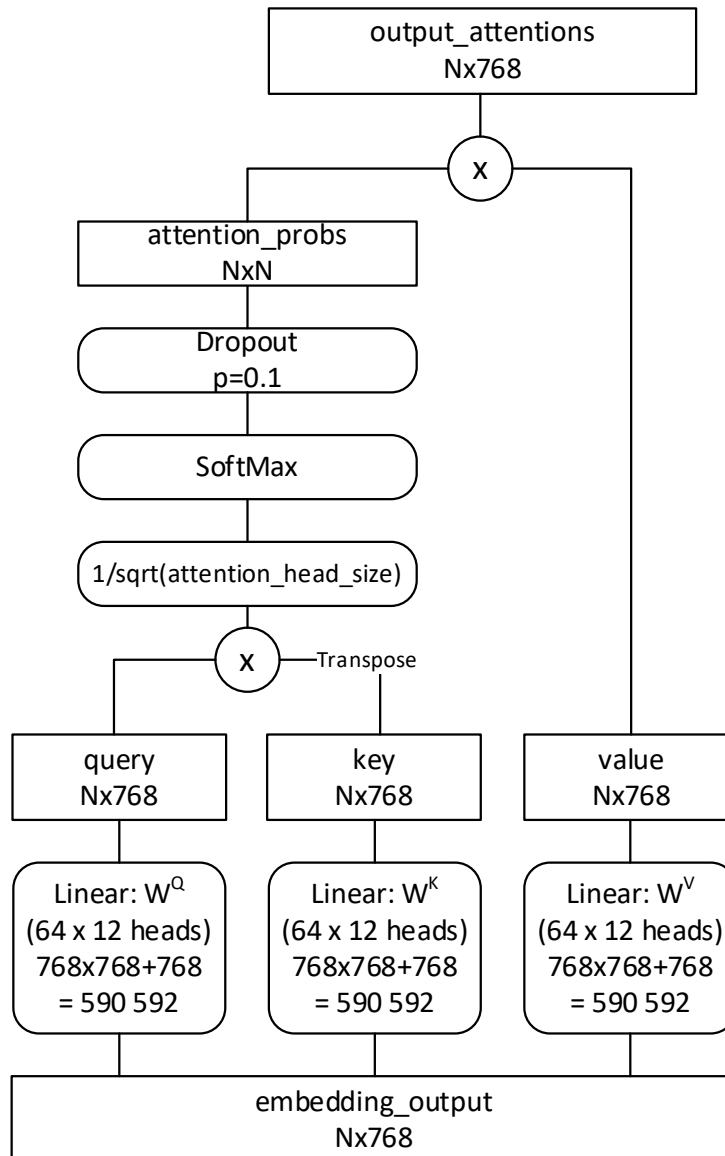
“Here’s the information I have!”

$$\text{Attention}(\underset{\substack{\text{from} \\ \text{blue}}}{q}, \underset{\substack{\text{to} \\ \text{yellow}}}{k}, v) = \overbrace{\text{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right)}^{\text{Attention weights}} v$$

vector dimensionality of K, V



Self-Attention



BertSelfAttention

hidden_size = 768
 num_attention_heads = 12
 attention_head_size = 64

$$\text{Attention}(Q, K, V) =$$

$$= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Self-Attention

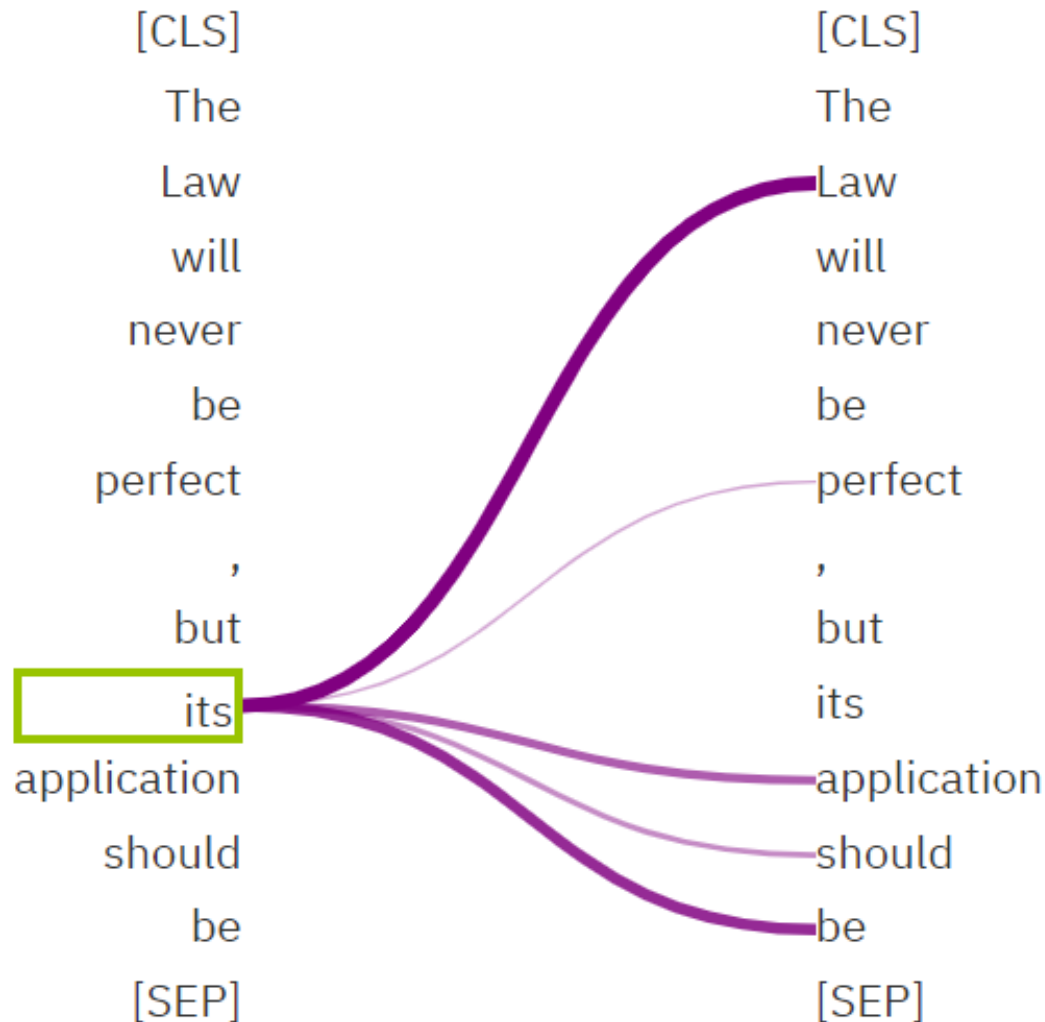
$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Токен	Q	K	V	QK^T	$QK^T/\sqrt{d_k}$	$softmax$	$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$	sum
Я	q_1	k_1	v_1	$q_1 k_1^T$	$q_1 k_1^T / 8$	a_{11}	$a_{11} v_1$	z_1
иду		k_2	v_2	$q_1 k_2^T$	$q_1 k_2^T / 8$	a_{12}	$a_{12} v_2$	
домой		k_3	v_3	$q_1 k_3^T$	$q_1 k_3^T / 8$	a_{13}	$a_{13} v_3$	

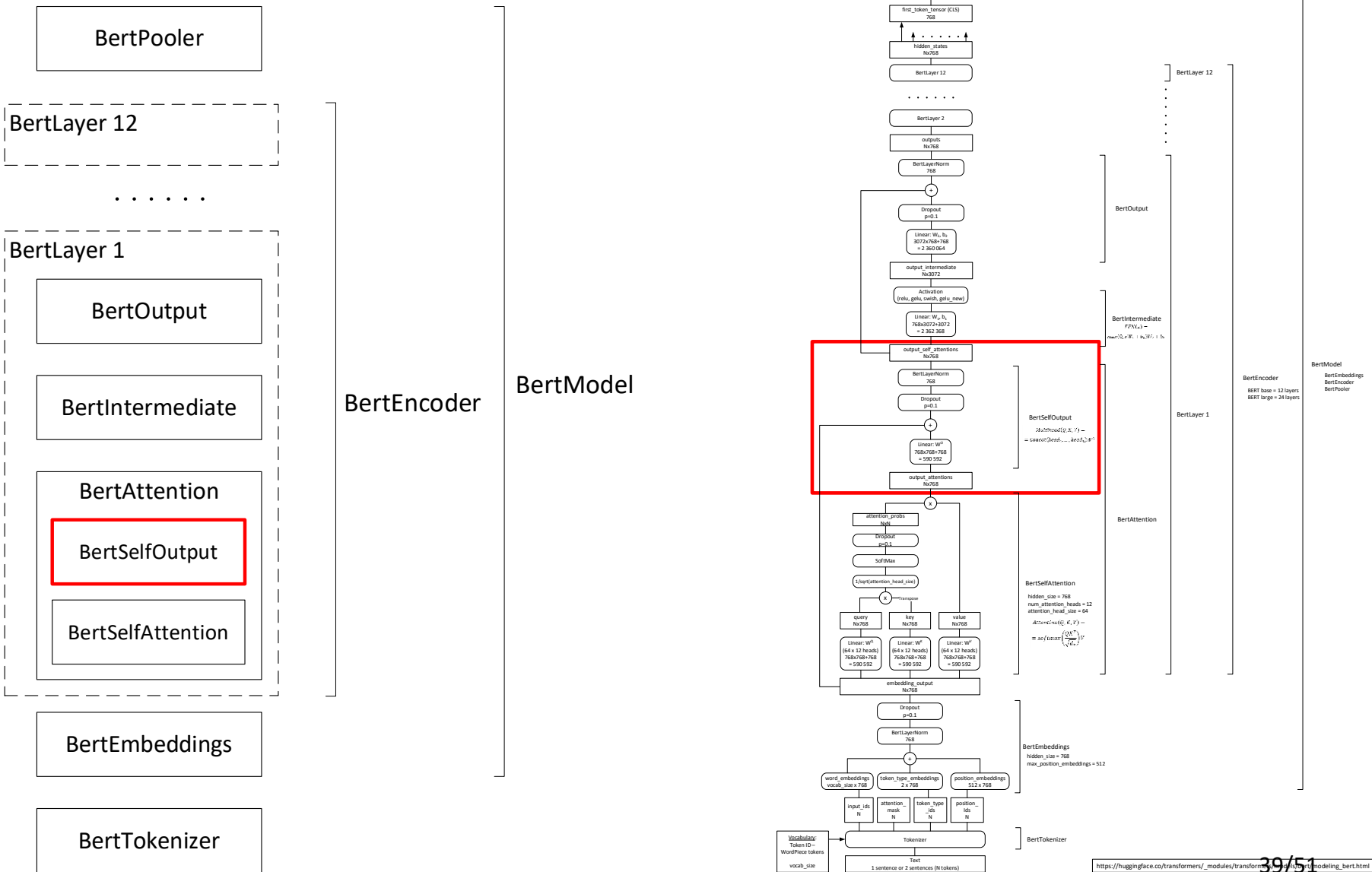
Токен	Q	K	V	QK^T	$QK^T/\sqrt{d_k}$	$softmax$	$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$	sum
Я		k_1	v_1	$q_2 k_1^T$	$q_2 k_1^T / 8$	a_{21}	$a_{21} v_1$	z_2
иду	q_2	k_2	v_2	$q_2 k_2^T$	$q_2 k_2^T / 8$	a_{22}	$a_{22} v_2$	
домой		k_3	v_3	$q_2 k_3^T$	$q_2 k_3^T / 8$	a_{23}	$a_{23} v_3$	

Self-Attention

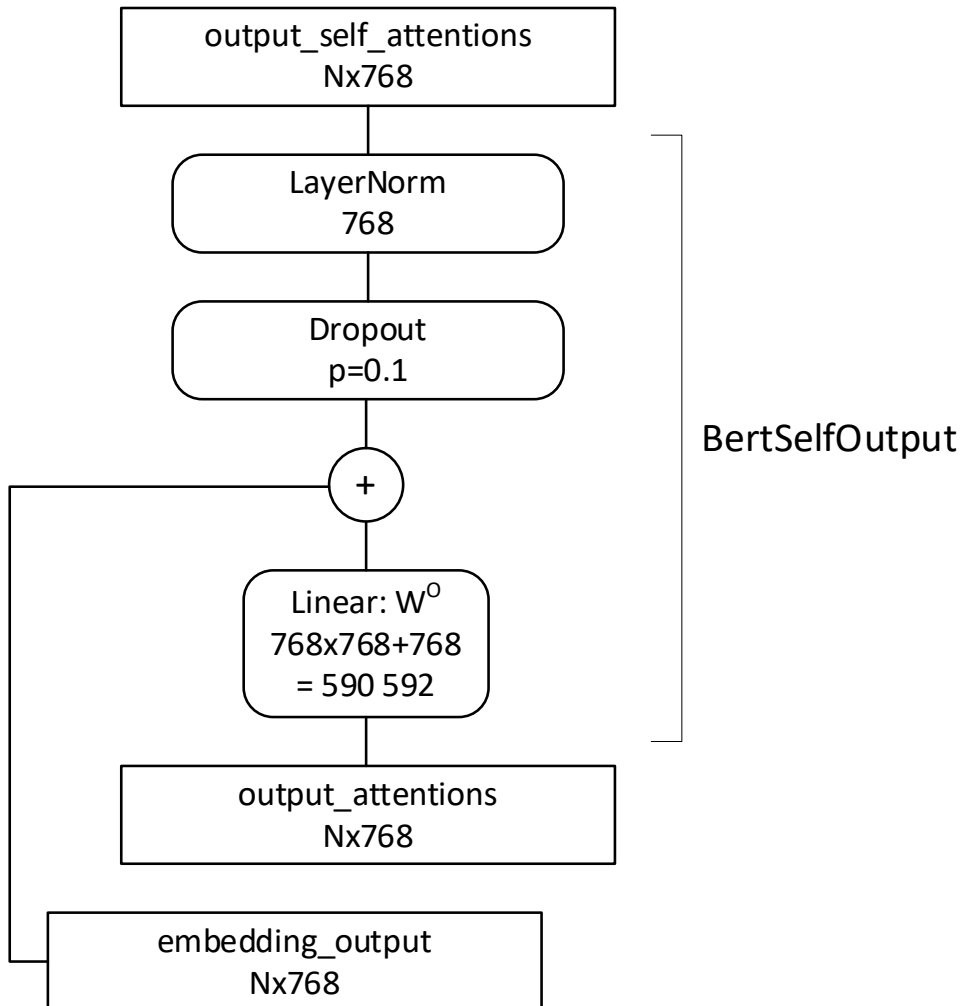
<https://huggingface.co/exbert>



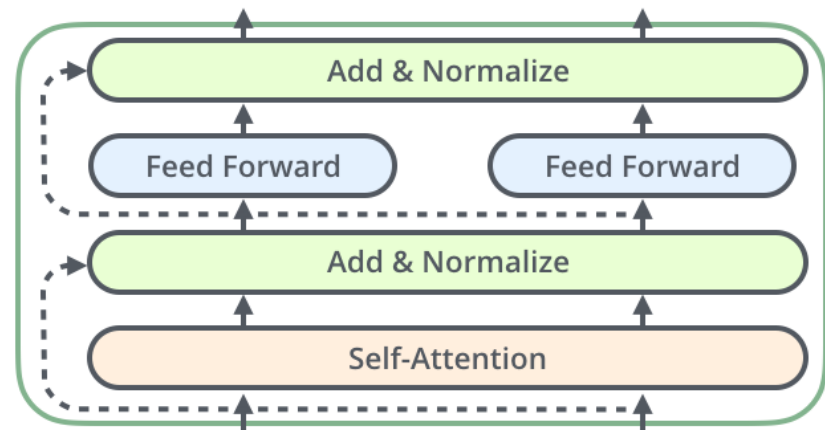
BertSelfOutput



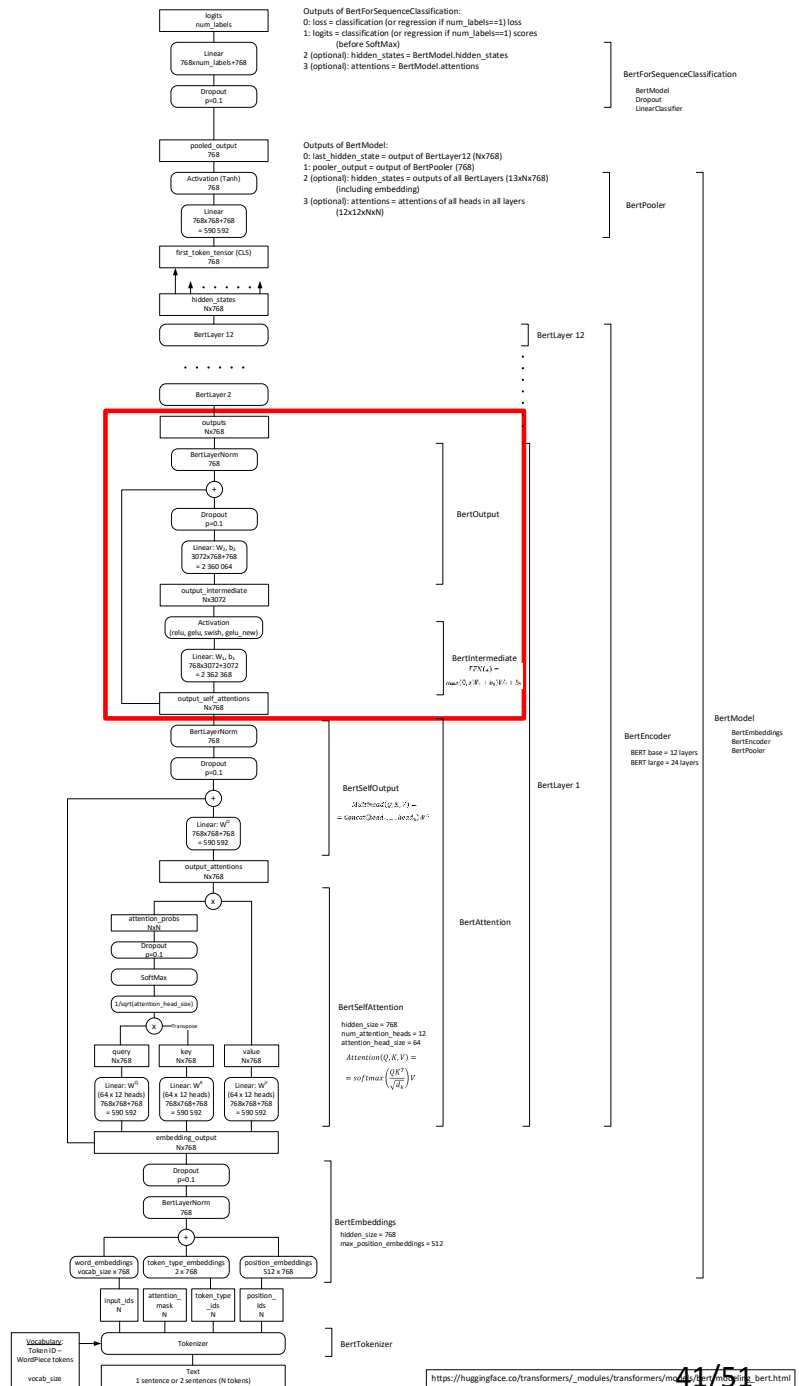
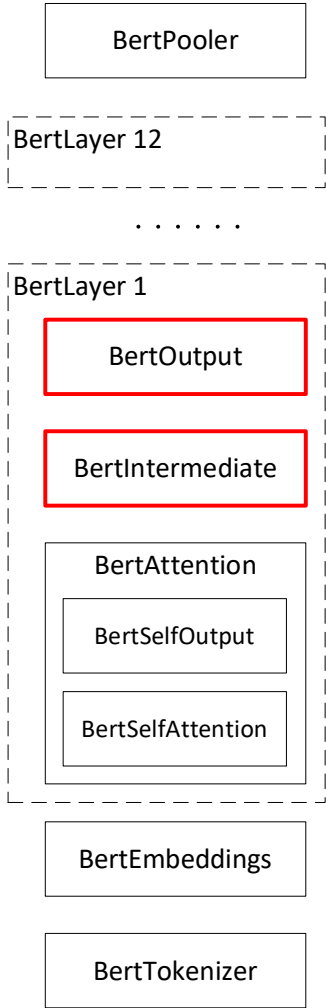
BertSelfOutput



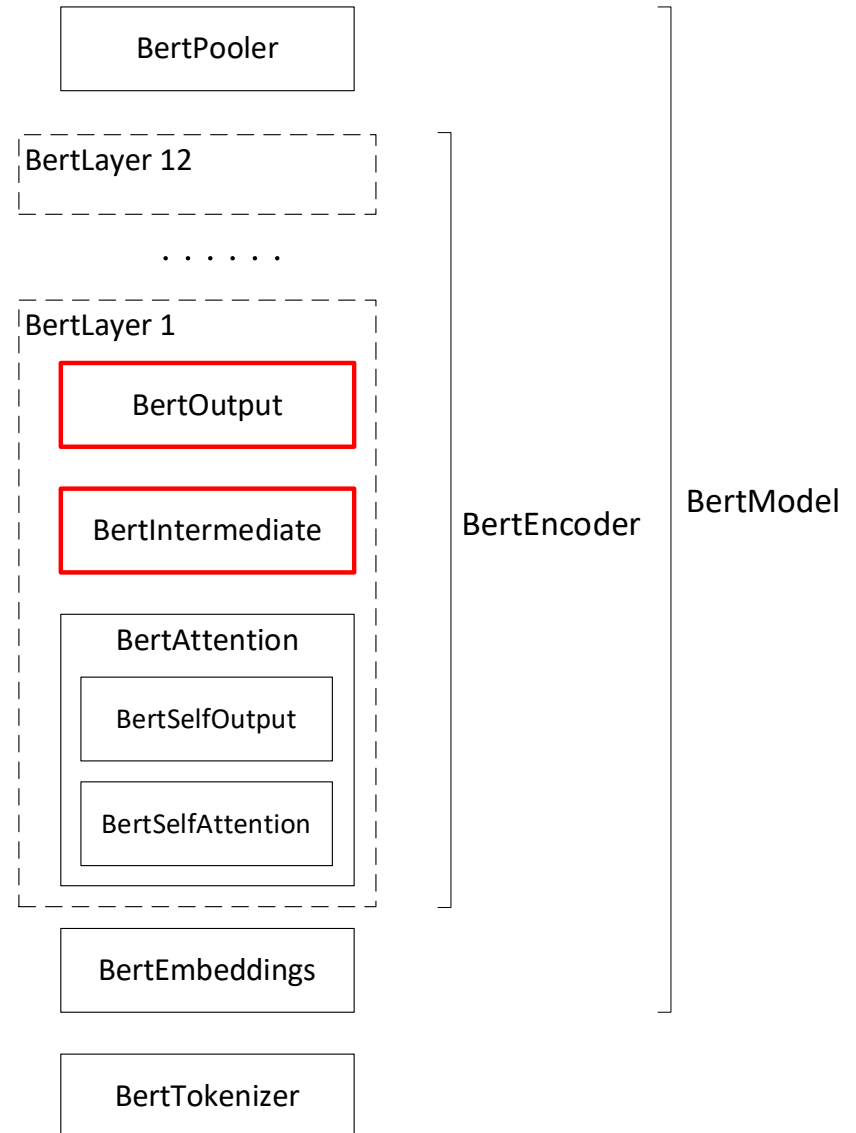
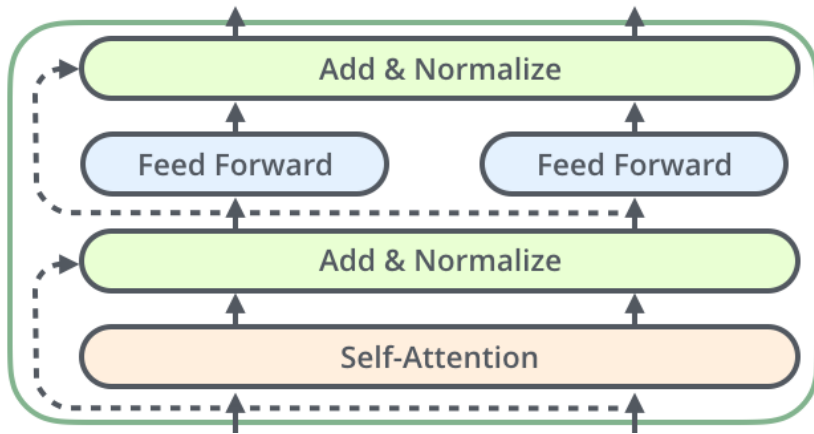
$$\text{Multihead}(Q, K, V) = \\ = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$



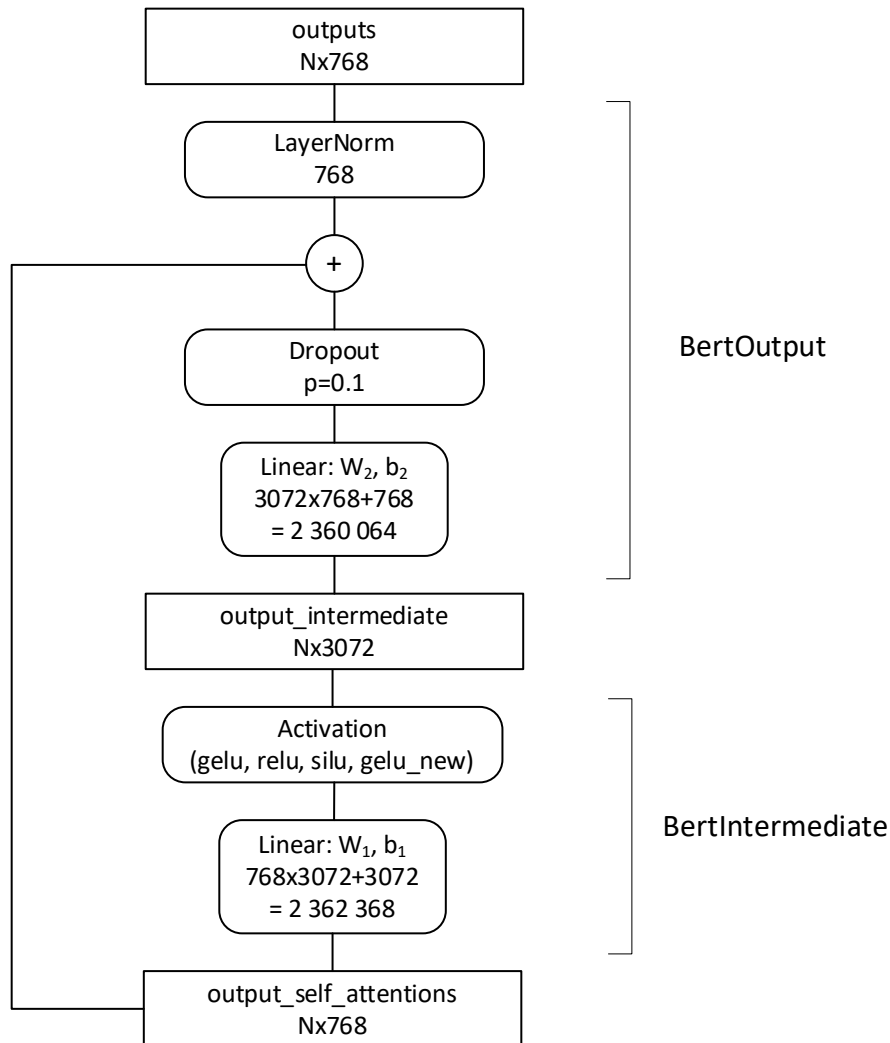
+ BertOutput



BertIntermediate + BertOutput

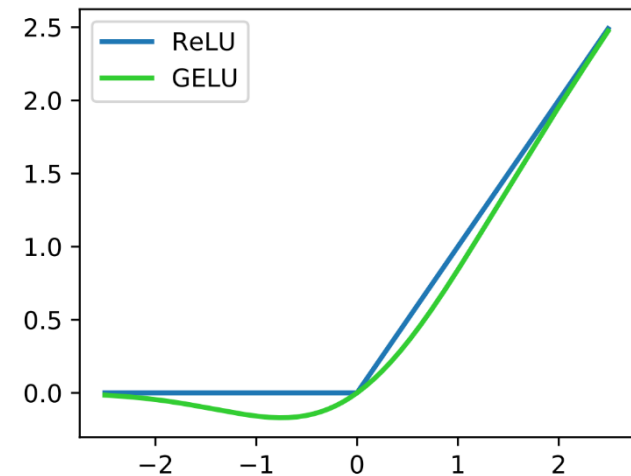


BertIntermediate + BertOutput



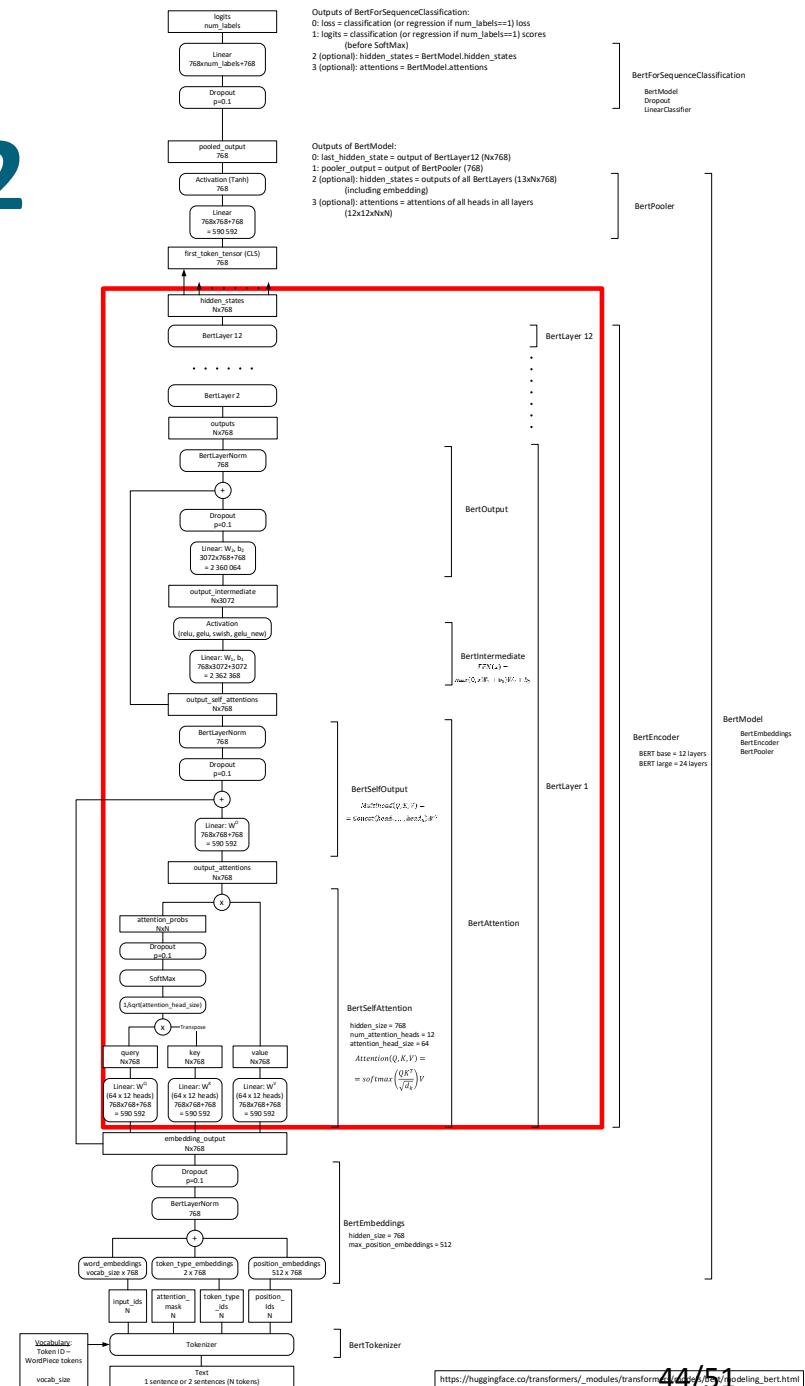
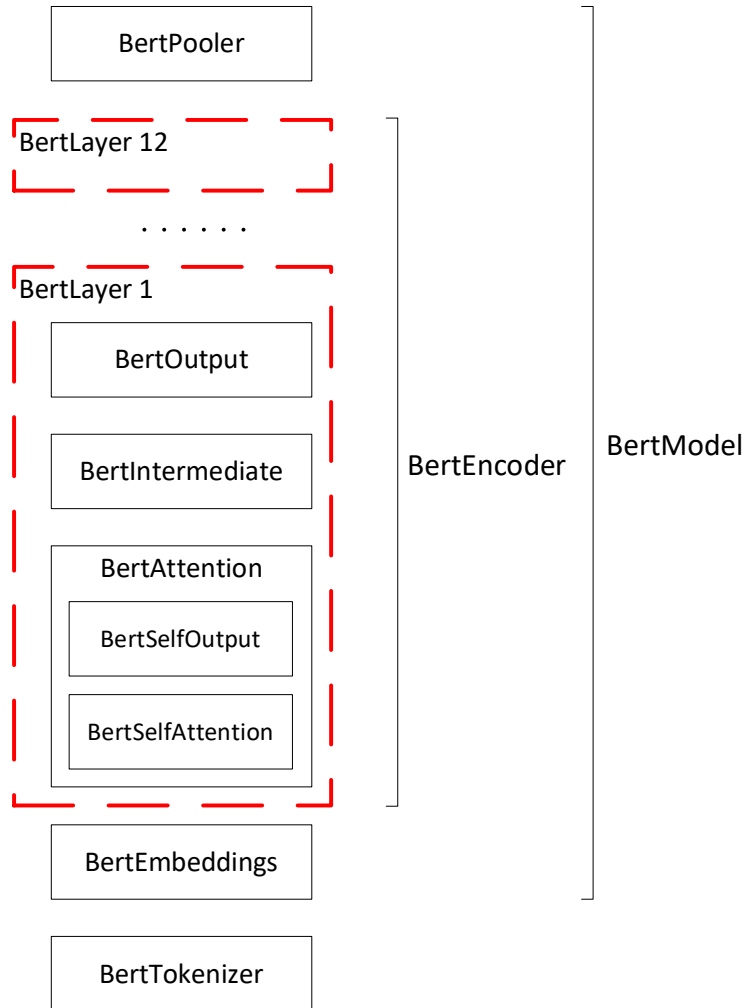
$$FFN(x) = \\ = \max(0, xW_1 + b_1)W_2 + b_2$$

GELU – Gaussian-error linear unit

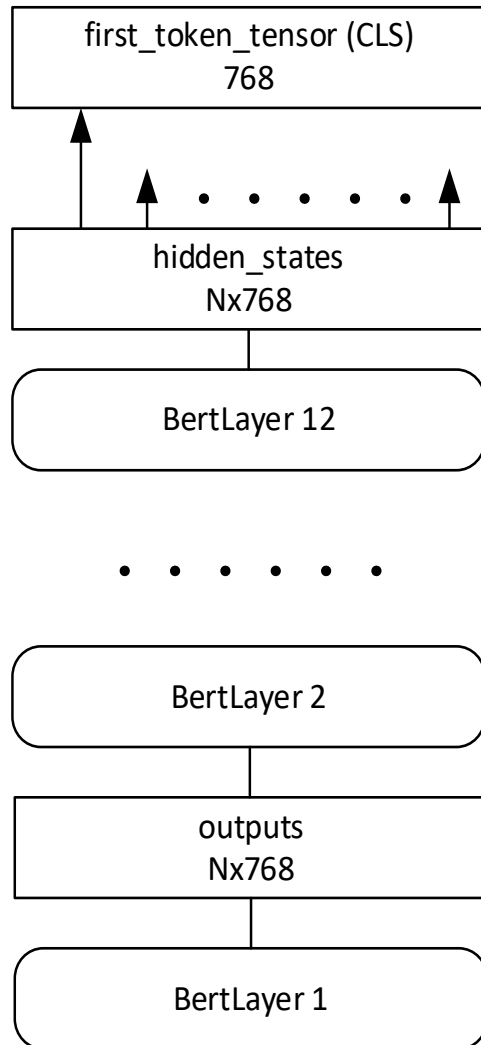


<https://github.com/huggingface/transformers/blob/main/src/transformers/activations.py>

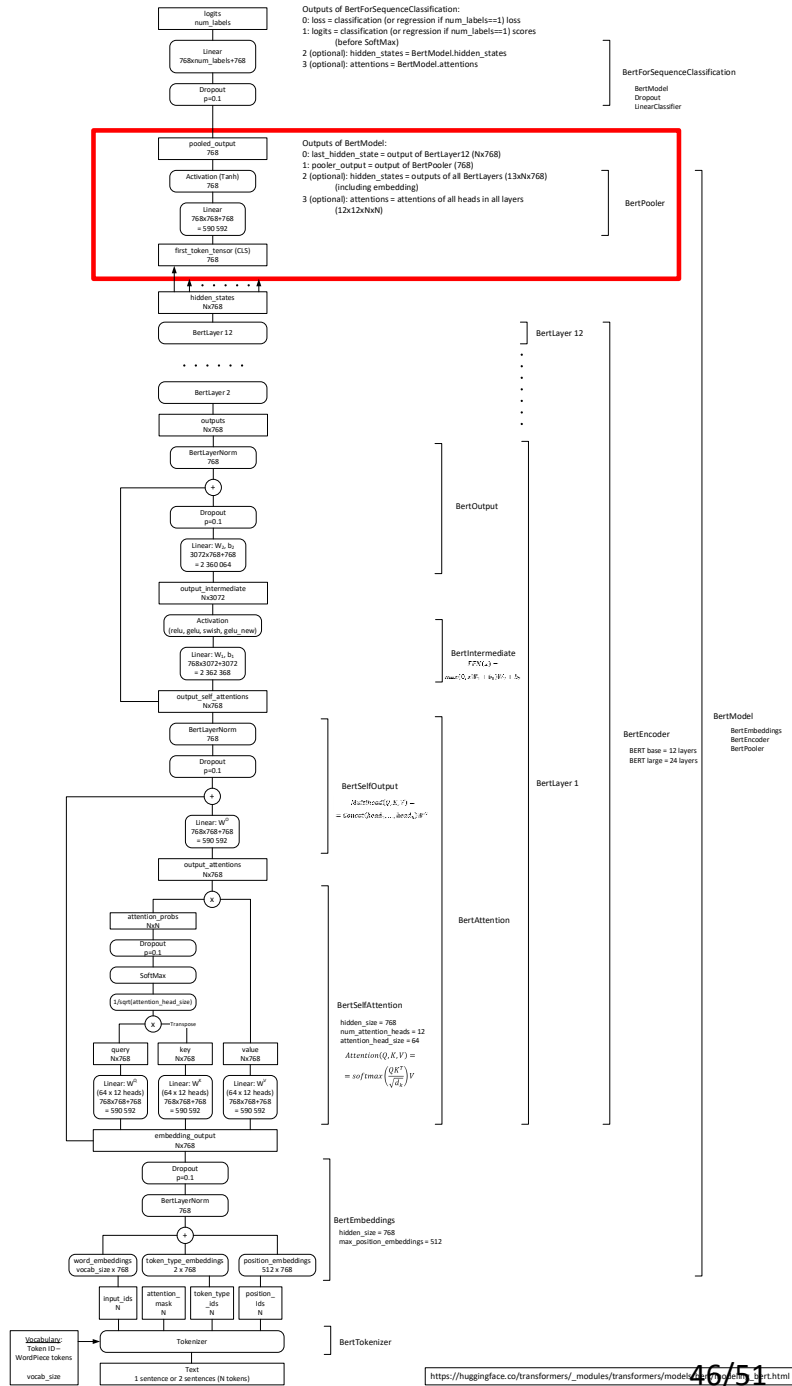
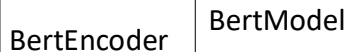
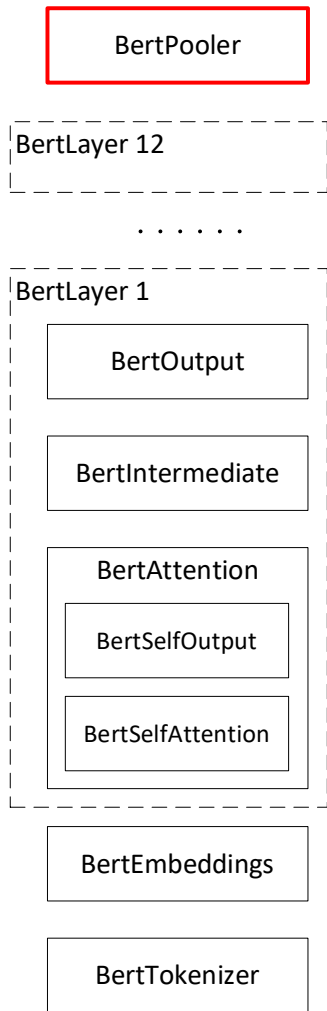
BertEncoder = BertLayer1...BertLayer12



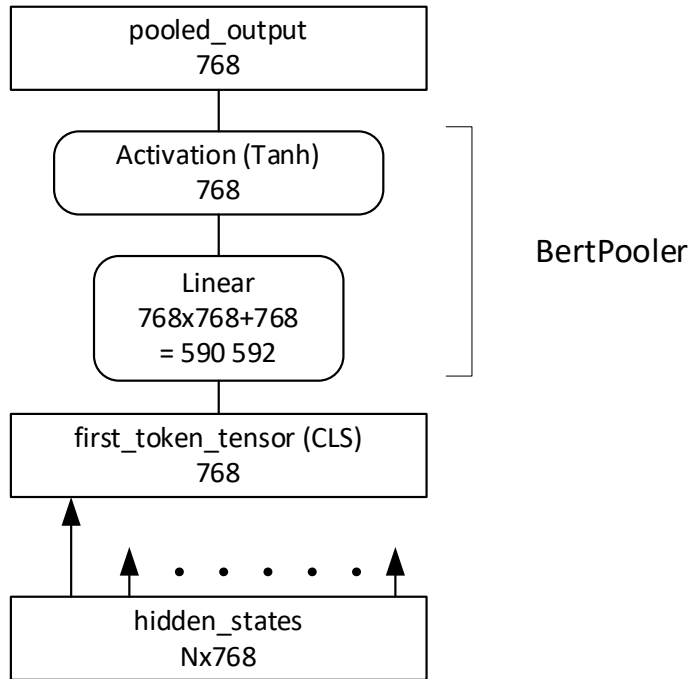
BertEncoder = BertLayer1...BertLayer12



BertPooler



BertPooler



Outputs of BertModel:

0: last_hidden_state = output of BertLayer12 (Nx768)

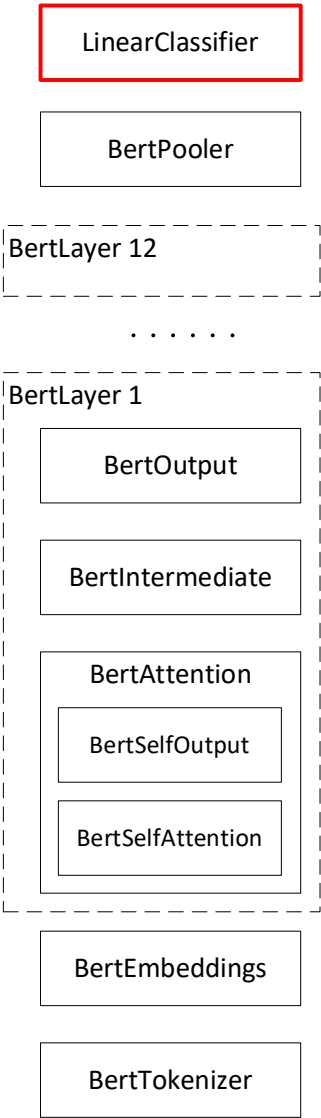
1: pooler_output = output of BertPooler (768)

2 (optional): hidden_states = outputs of all BertLayers (13xNx768)
(including embedding)

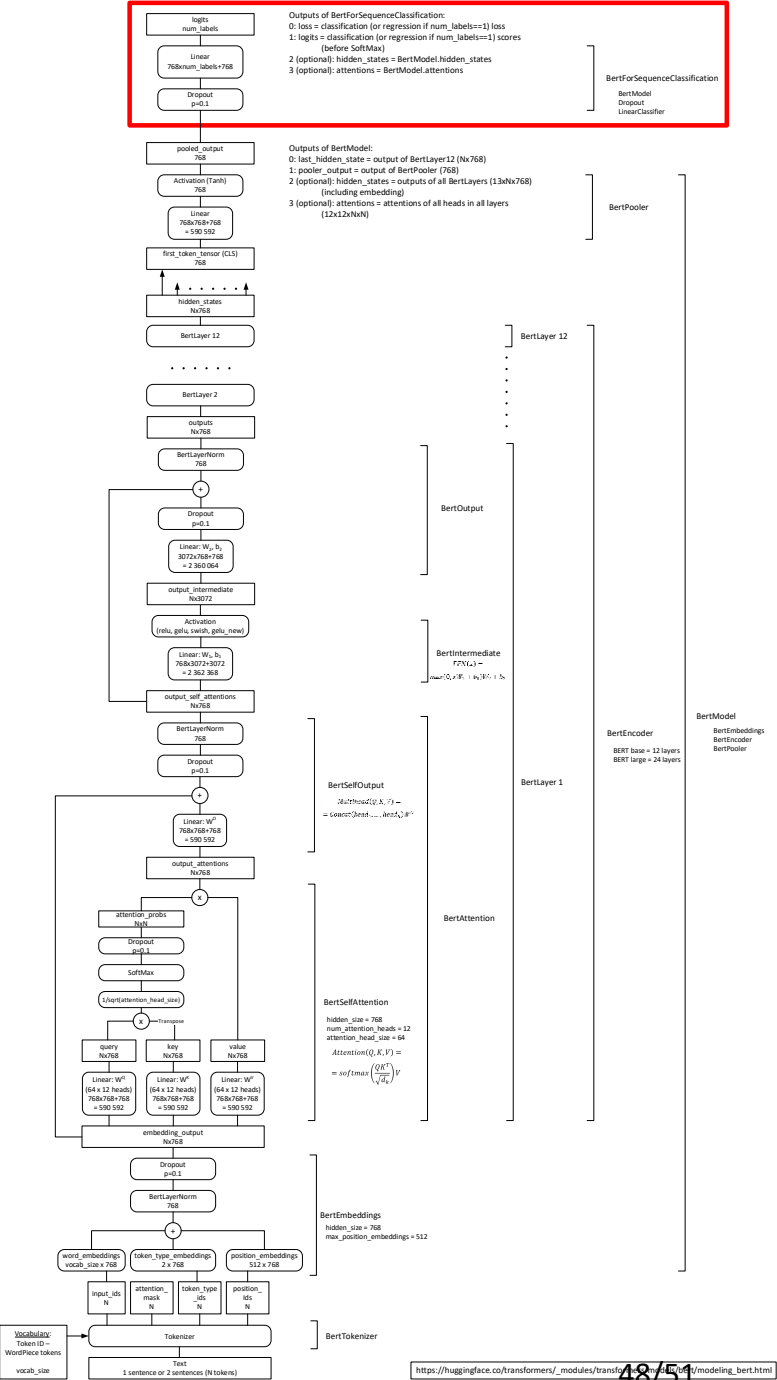
3 (optional): attentions = attentions of all heads in all layers
(12x12xNxN)

Этот слой обучается решать задачу
Next Sentence Prediction
(в процессе обучения BERT)

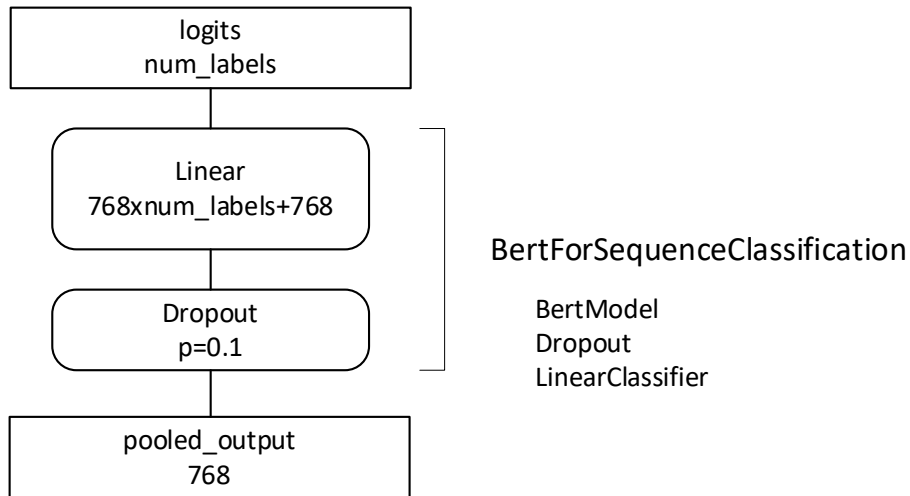
BertForSequenceClassification: LinearClassifier



BertEncoder BertModel



BertForSequenceClassification: LinearClassifier



Outputs of BertForSequenceClassification:

0: loss = classification (or regression if $\text{num_labels}==1$) loss

1: logits = classification (or regression if $\text{num_labels}==1$) scores
(before SoftMax)

2 (optional): hidden_states = BertModel.hidden_states

3 (optional): attentions = BertModel.attentions

Hugging Face: transformers

- Main website:
 - <https://huggingface.co>
- Models:
 - <https://huggingface.co/models>
- Transformers:
 - <https://github.com/huggingface/transformers>
- Training and fine-tuning:
 - <https://huggingface.co/docs/transformers/training>
- BERT – documentation:
 - https://huggingface.co/docs/transformers/model_doc/bert
- Source code:
 - https://github.com/huggingface/transformers/blob/main/src/transformers/models/bert/modeling_bert.py

Hugging Face: модели

- Русскоязычные модели (включая многоязычные):
 - <https://huggingface.co/models?filter=ru>
- Модели DeepPavlov
 - <https://huggingface.co/DeepPavlov>
- Модели Сбер
 - <https://huggingface.co/ai-forever>
- См. демонстрационный ноутбук

RoBERTa

RoBERTa

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach
 - <https://arxiv.org/abs/1907.11692>
 - Facebook AI*
 - Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle

Вклад

- Воспроизведение обучения BERT
- Оценка влияния гиперпараметров и размера набора данных
- BERT был значительно недообучен
- SOTA на GLUE, RACE и SQuAD

Выводы

- Требуется более длительное обучение модели с большими батчами и на большем объеме данных
- Задачу Next Sentence Prediction можно исключить
- Требуется обучение на более длинных последовательностях
- Динамическое изменение масок для обучающих данных – токены маскируются по-разному на разных эпохах (в BERT однократно)

Обучение (BERT → RoBERTa)

- Adam:
 - $\beta_1 = 0.9$
 - $\beta_2 = 0.999 \rightarrow 0.98$
 - $\epsilon = 10^{-6}$
 - Weight decay $L_2 = 0.01$
- Learning rate:
 - warm-up steps: 10 000
 - $\max = 10^{-4}$ (linearly decayed)
- Dropout: 0.1
- Activation function: GELU
- Размер батча: 256 (128 000 токенов) → 8 000
- Количество шагов: 1 000 000 → 500 000
- Количество эпох: 40

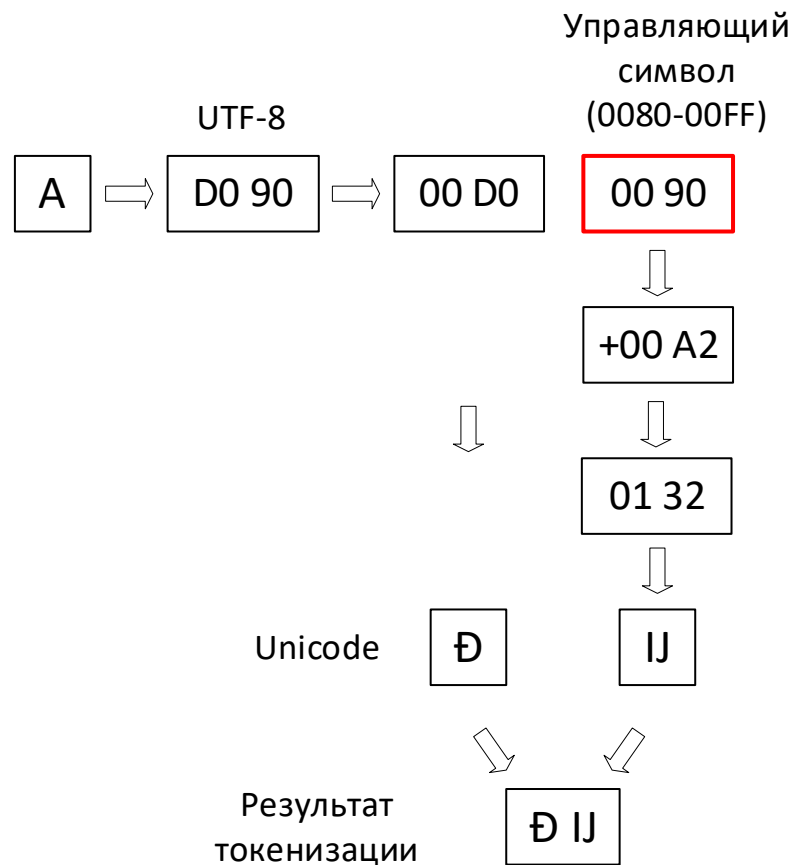
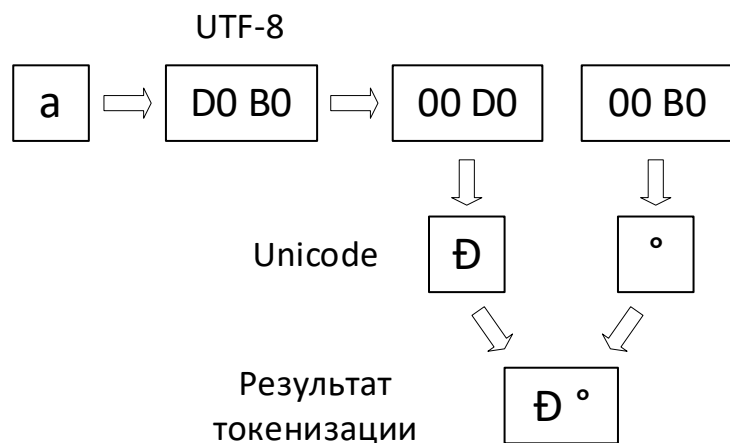
DGX-1: 8 × 32GB Nvidia V100 GPUs
interconnected by InfiniBand

Datasets: 3.3 млрд токенов,
16 ГБ → 160 ГБ

Токенизация

- BERT: WordPiece, 30 000 токенов
- RoBERTa: Byte-level BPE, 50 000 токенов
 - Как в GPT-2 [Radford et al., 2019]
 - Кодировается любое слово на любом языке
 - “Early experiments revealed only slight differences between these encodings, with the Radford et al. (2019) BPE achieving slightly worse end-task performance on some tasks. Nevertheless, we believe the advantages of a universal encoding scheme outweighs the minor degradation in performance and use this encoding in the remainder of our experiments.”

Токенизация



Символ	Unicode		UTF-8		CP-1251
	Hex	Dec	Hex	Dec	
А	0410	1040	D090	208 144	192
Б	0411	1041	D091	208 145	193
В	0412	1042	D092	208 146	194
...					
а	0430	1072	D0B0	208 176	224
б	0431	1073	D0B1	208 177	225
в	0432	1074	D0B2	208 178	226
...					

GLUE Benchmark

- GLUE (General Language Understanding Evaluation) benchmark – платформа для оценки систем понимания естественного языка
 - <https://gluebenchmark.com>
- SuperGLUE
 - <https://super.gluebenchmark.com>

GLUE Benchmark

1. The Corpus of Linguistic Acceptability (CoLA)
2. The Stanford Sentiment Treebank (SST-2)
3. Microsoft Research Paraphrase Corpus (MRPC)
4. Semantic Textual Similarity Benchmark (STS-B)
5. Quora Question Pairs (QQP)
6. MultiNLI Matched (MNLI-m)
7. MultiNLI Mismatched (MNLI-mm)
8. Question NLI (QNLI, SQuAD)
9. Recognizing Textual Entailment (RTE)
10. Winograd NLI (WNLI)
11. Diagnostics Main (AX)

GLUE Benchmark

Rank Name		Model	URL	Score
1	Microsoft Alexander v-team	Turing ULR v6		91.3
2	JDExplore d-team	Vega v1		91.3
3	Microsoft Alexander v-team	Turing NLR v5		91.2
4	DIRL Team	DeBERTa + CLEVER		91.1
5	ERNIE Team - Baidu	ERNIE		91.1
6	AliceMind & DIRL	StructBERT + CLEVER		91.0
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8
8	HFL iFLYTEK	MacALBERT + DKM		90.7
9	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6
10	T5 Team - Google	T5		90.3

GLUE Benchmark

11	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9
12	Huawei Noah's Ark Lab	NEZHA-Large		89.8
13	LG AI Research	ANNA		89.8
14	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7
15	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4
16	David Kim	2digit LANet		89.3
17	倪仕文	DropAttack-RoBERTa-large		88.8
18	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4
19	Junjie Yang	HIRE-RoBERTa		88.3
20	Shiwen Ni	ELECTRA-large-M (bert4keras)		88.3
21	Facebook AI	RoBERTa		88.1

Динамическое маскирование

Masking	SQuAD 2.0	MNLI-m	SST-2
reference	76.3	84.3	92.8
<i>Our reimplementation:</i>			
static	78.3	84.3	92.5
dynamic	78.7	84.0	92.9

Table 1: Comparison between static and dynamic masking for BERT_{BASE}. We report F1 for SQuAD and accuracy for MNLI-m and SST-2. Reported results are medians over 5 random initializations (seeds). Reference results are from [Yang et al. \(2019\)](#).

Исключение NSP

Model	SQuAD 1.1/2.0	MNLI-m	SST-2	RACE
<i>Our reimplementation (with NSP loss):</i>				
SEGMENT-PAIR	90.4/78.7	84.0	92.9	64.2
SENTENCE-PAIR	88.7/76.2	82.9	92.1	63.0
<i>Our reimplementation (without NSP loss):</i>				
FULL-SENTENCES	90.4/79.1	84.7	92.5	64.8
DOC-SENTENCES	90.6/79.7	84.7	92.7	65.6
BERT _{BASE}	88.5/76.3	84.3	92.8	64.3
XLNet _{BASE} (K = 7)	-/81.3	85.8	92.7	66.1
XLNet _{BASE} (K = 6)	-/81.0	85.6	93.4	66.7

Без пересечения границ документов (переменный размер батча)

Размер батча и скорость обучения

bsz	steps	lr	ppl	MNLI-m	SST-2
256	1M	1e-4	3.99	84.7	92.7
2K	125K	7e-4	3.68	85.2	92.9
8K	31K	1e-3	3.77	84.6	92.8

Table 3: Perplexity on held-out training data (*ppl*) and development set accuracy for base models trained over BOOKCORPUS and WIKIPEDIA with varying batch sizes (*bsz*). We tune the learning rate (*lr*) for each setting. Models make the same number of passes over the data (epochs) and have the same computational cost.

Больше данных и дольше обучение

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	94.6/89.4	90.2	96.4
BERT _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
XLNet _{LARGE}						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

Table 4: Development set results for RoBERTa as we pretrain over more data (16GB $\rightarrow$ 160GB of text) and pretrain for longer (100K $\rightarrow$ 300K $\rightarrow$ 500K steps). Each row accumulates improvements from the rows above. RoBERTa matches the architecture and training objective of BERT_{LARGE}. Results for BERT_{LARGE} and XLNet_{LARGE} are from Devlin et al. (2019) and Yang et al. (2019), respectively. Complete results on all GLUE tasks can be found in the Appendix.

Результаты на GLUE

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
<i>Single-task single models on dev</i>										
BERT _{LARGE}	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet _{LARGE}	89.8/-	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa	90.2/90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	91.3	-
<i>Ensembles on test (from leaderboard as of July 25, 2019)</i>										
ALICE	88.2/87.9	95.7	90.7	83.5	95.2	92.6	68.6	91.1	80.8	86.3
MT-DNN	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2/89.8	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5

Table 5: Results on GLUE. All results are based on a 24-layer architecture. BERT_{LARGE} and XLNet_{LARGE} results are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively. RoBERTa results on the development set are a median over five runs. RoBERTa results on the test set are ensembles of *single-task* models. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model. Averages are obtained from the GLUE leaderboard.

Итоговая модель RoBERTa

- RoBERTa = Robustly optimized BERT approach
- Динамическое маскирование
- Вход – по предложениям (Full Sentences)
 - с вхождением предложений из разных документов
- Большой размер батча (8 000)
- Byte-level BPE (50 257 токенов)

Модели

- RoBERTa-base – 12 слоев, 110 млн параметров
 - <https://huggingface.co/roberta-base>
- RoBERTa-large – 24 слоя, 355 млн параметров
 - <https://huggingface.co/roberta-large>
 - 1024 V100 – 1 день
- ruRoBERTa-large – 24 слоя, 355 млн параметров
 - <https://huggingface.co/ai-forever/ruRoberta-large>
 - <https://habr.com/ru/companies/sberbank/articles/567776/>
 - 64 V100 – 3 недели

GPT

– Generative Pre-trained
Transformer

План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

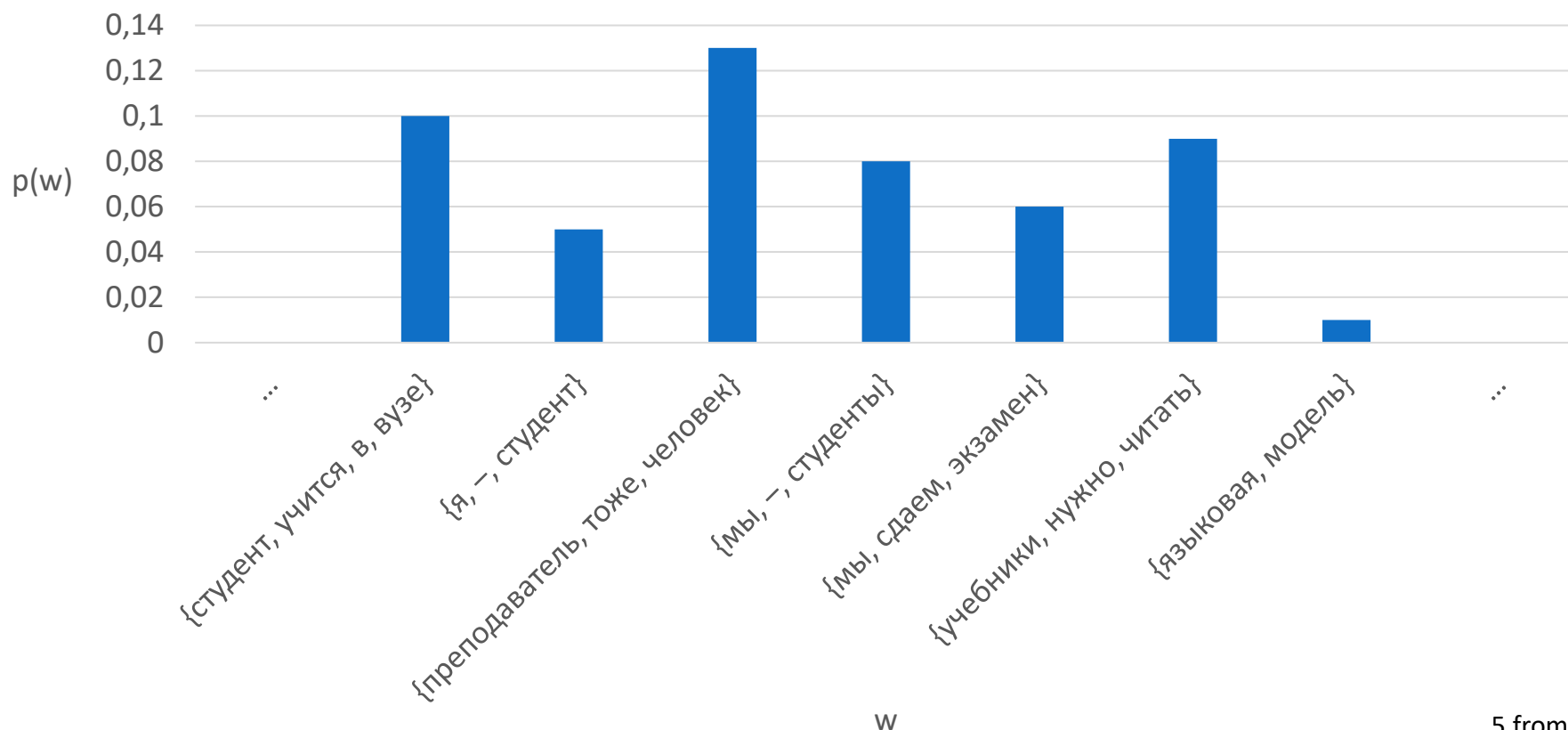
Языковые модели (1)

- **Языковая модель** (Language Model) – алгоритм, моделирующий вероятностное распределение над различными лингвистическими единицами (символами, словами, предложениями, ...)
- Пример: совместное распределение слов

$$P(w_1, \dots, w_m)$$

Языковые модели (2)

- Основная функция языковой модели – моделирование вероятностного распределения последовательностей:



Языковые модели (3)

- В общем случае, для i -го слова:

$$P(w_i | context)$$

- Предыдущие слова (left-to-right, autoregressive model, causal LM):

$$P(w_i | w_{i-k}, \dots, w_{i-1})$$

- Двухнаправленный контекст (bidirectional, autoencoding model):

$$P(w_i | w_{i-k}, \dots, w_{i-1}, w_{i+1}, w_{i+k})$$

- Совместное распределение (цепное правило):

$$P(w_1, \dots, w_m) = \prod_{i=1}^m P(w_i | w_{<i})$$

Языковые модели (4)



кто придумал

кто придумал **поп ит**

кто придумал **интернет**

кто придумал **баскетбол**

кто придумал **школу**

кто придумал **водку**

кто придумал **шаурму**

кто придумал **пельмени**

кто придумал **телефон**

кто придумал **домашнее задание**

кто придумал **пиво**

Яндекс

кто придумал

кто придумал коронавирус

кто придумал коронавирус и зачем

кто придумал интернет

кто придумал телефон

кто придумал футбол

кто придумал алфавит

кто придумал школу → [Платон и Аристотель](#)

кто придумал что тебя я не люблю

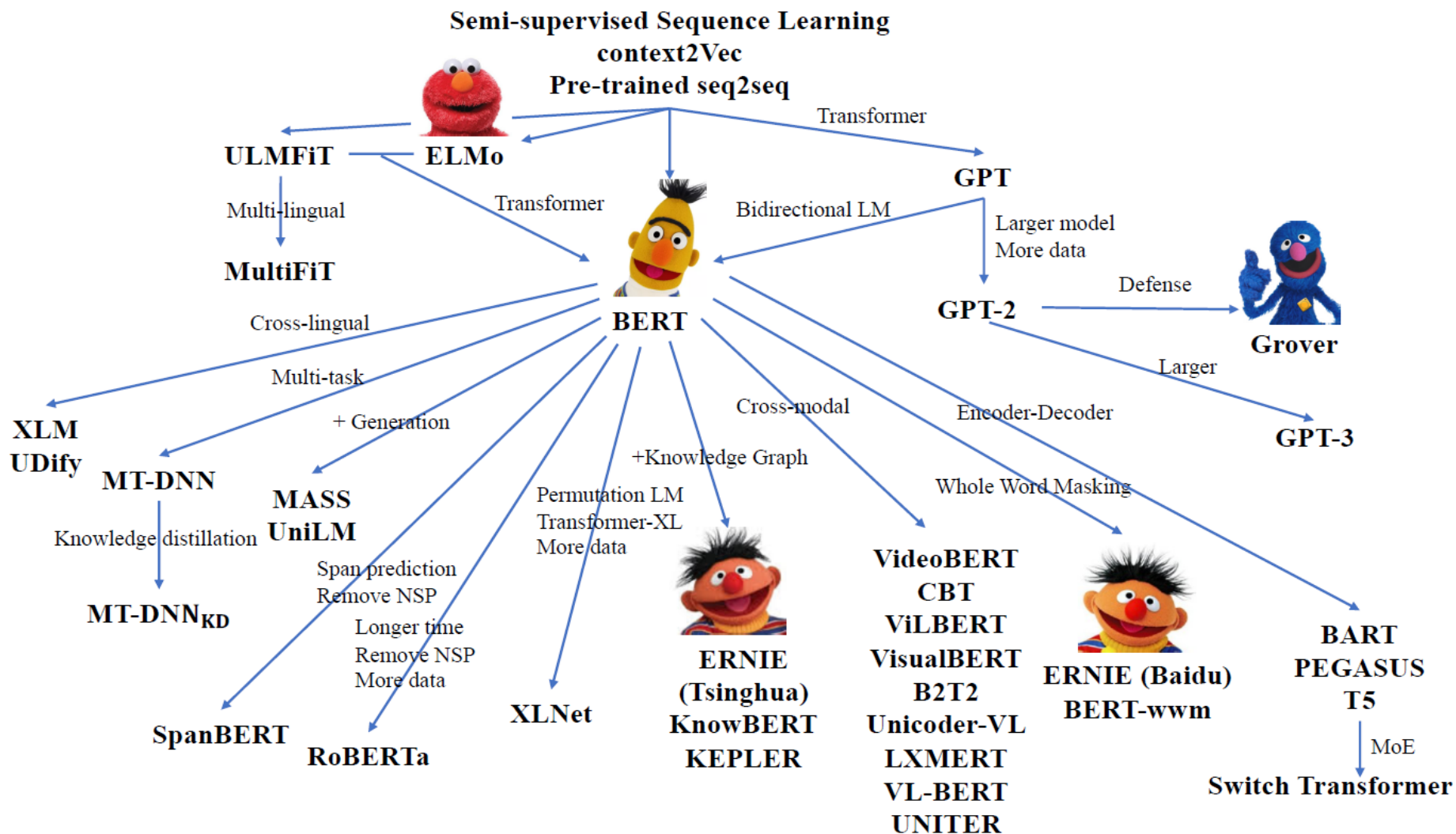
кто придумал домашнее задание в школе

кто придумал домашнее задание

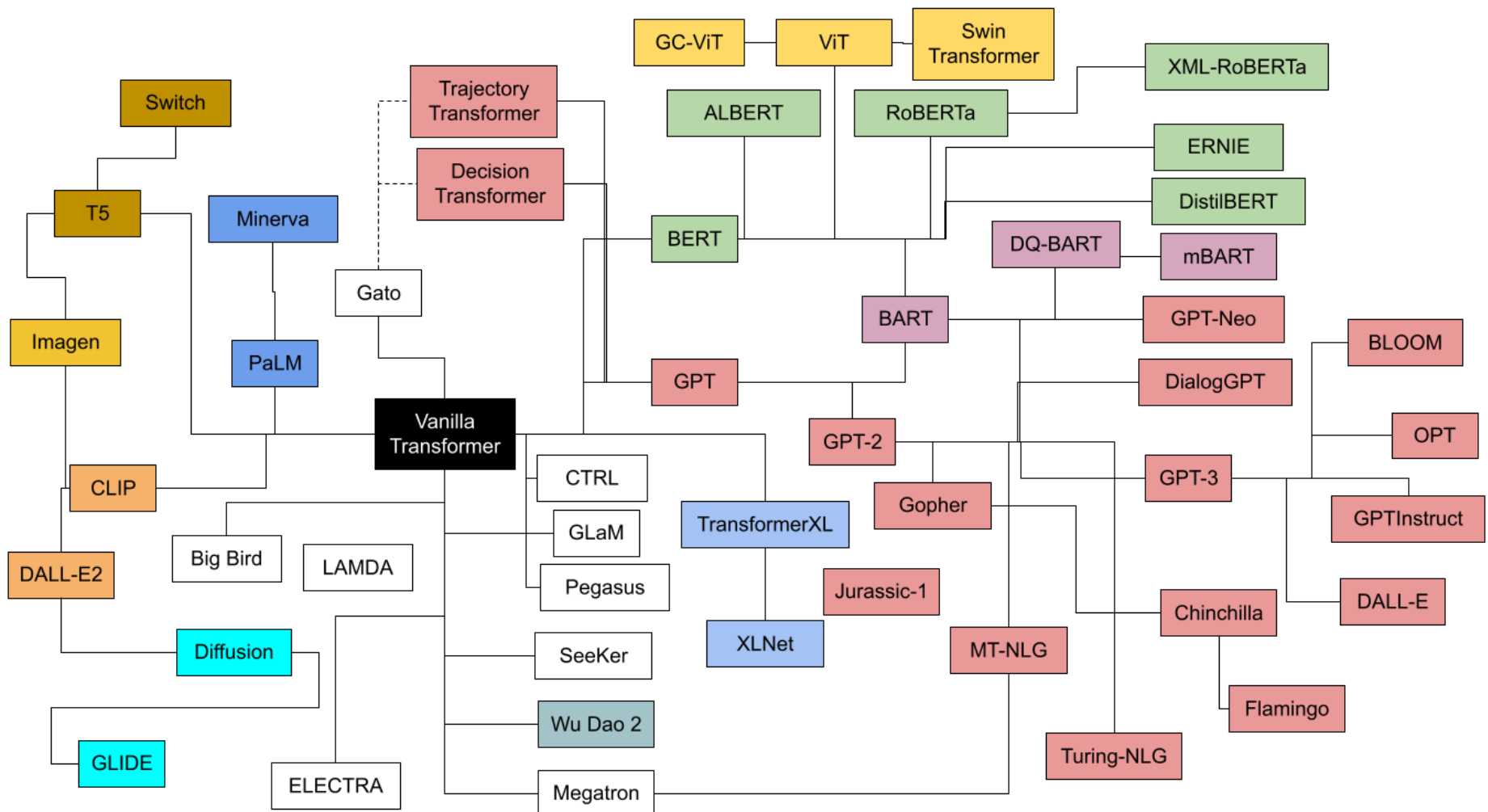
Языковые модели (5)

- Статистические языковые модели
 - N-граммные
 - Скрытые марковские модели
- Нейросетевые языковые модели
 - Полносвязные неглубокие (word2vec, fastText)
 - Рекуррентные (RNN, LSTM, GRU)
 - На основе внимания (Transformer, BERT, GPT)

Языковые модели (6)

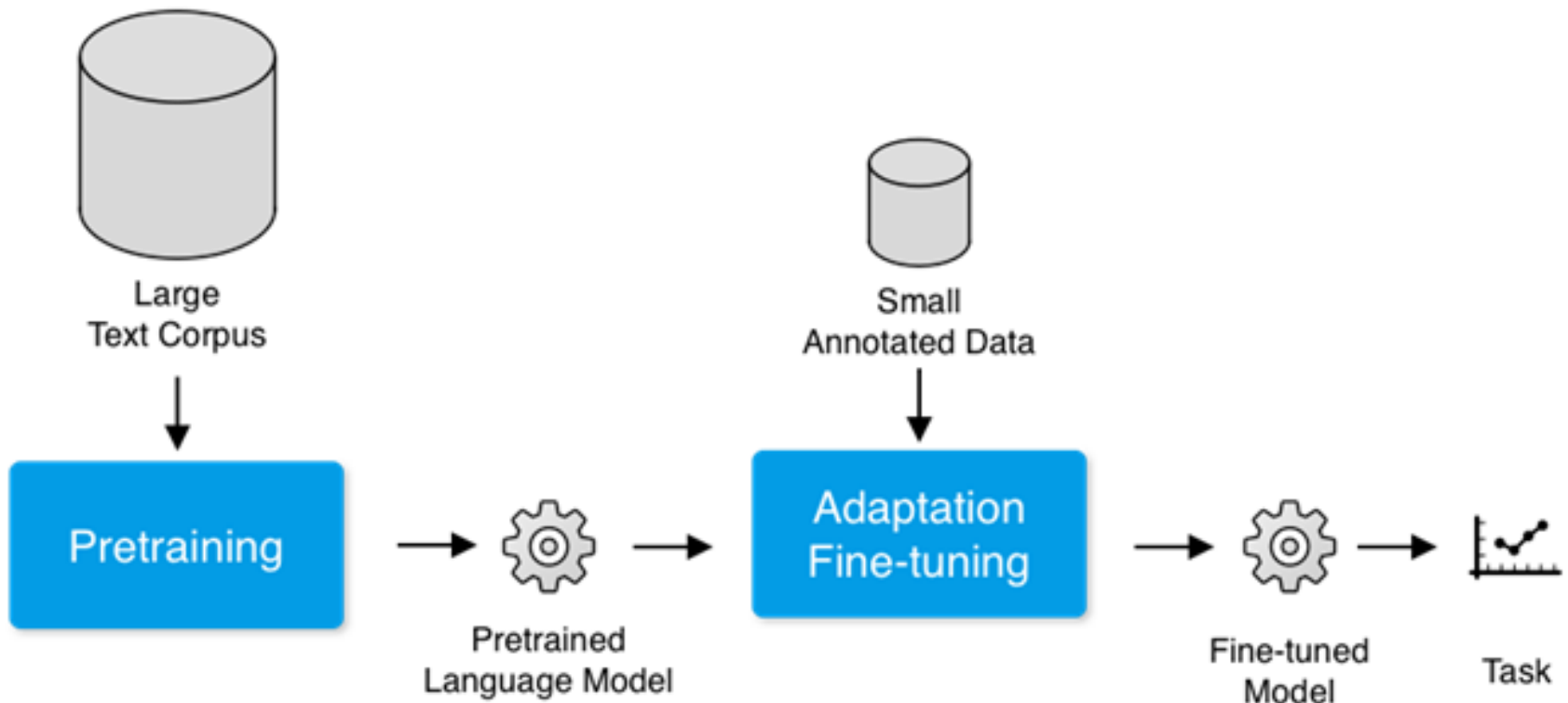


Языковые модели (7)



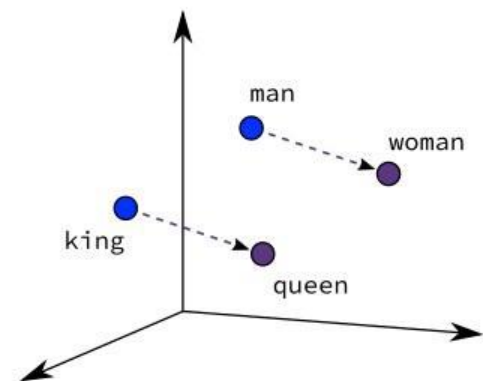
Языковые модели (8)

- Transfer Learning in NLP = Pre-Training and Fine-Tuning

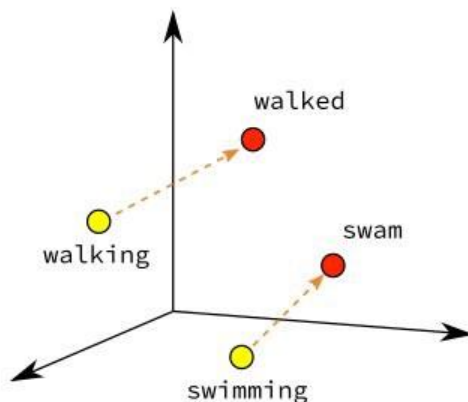


Языковые модели (9)

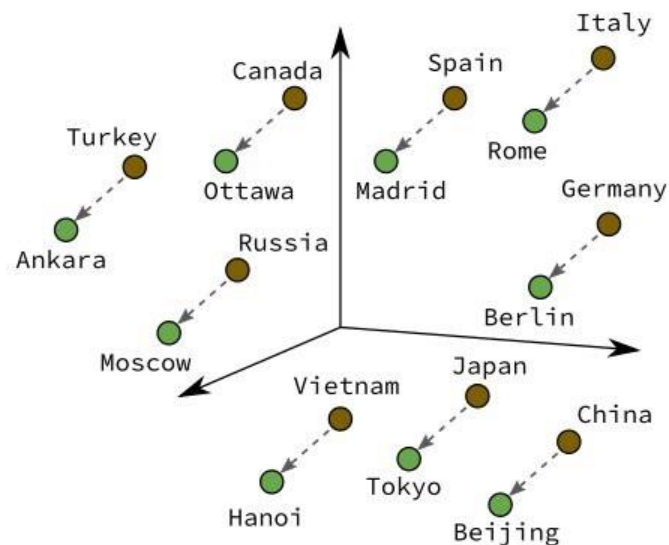
- Feature engineering → distributed representations (character/subword/word/phrase/sentence/document embeddings)



Male-Female



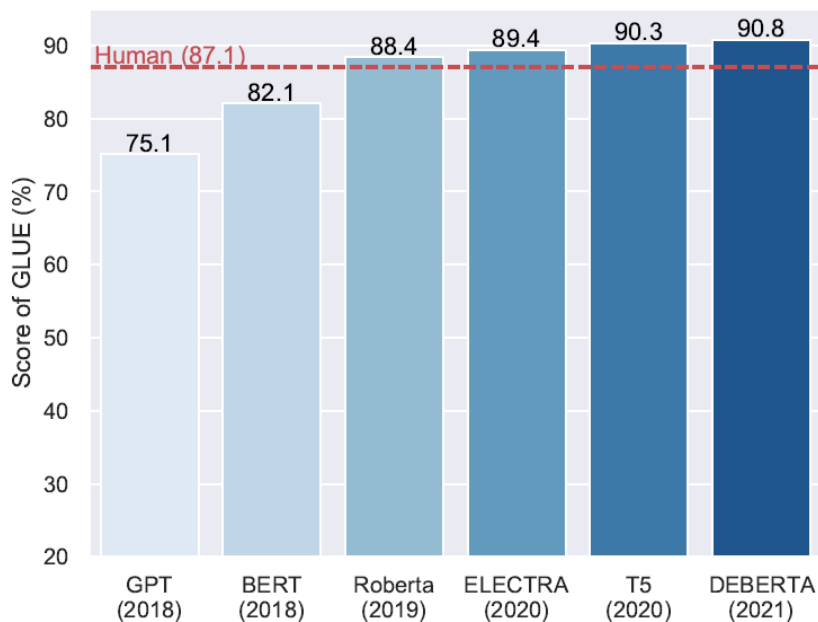
Verb Tense



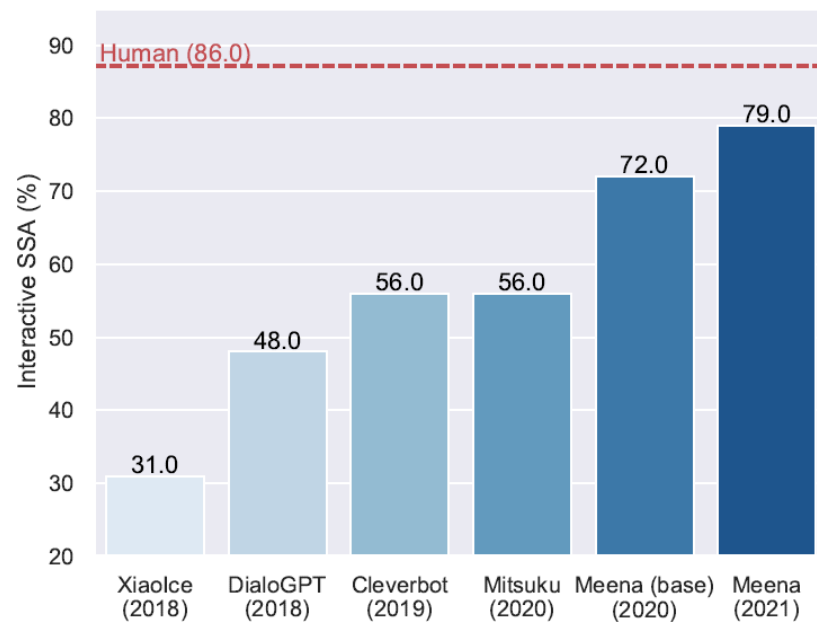
Country-Capital

Языковые модели (10)

GLUE, SuperGLUE

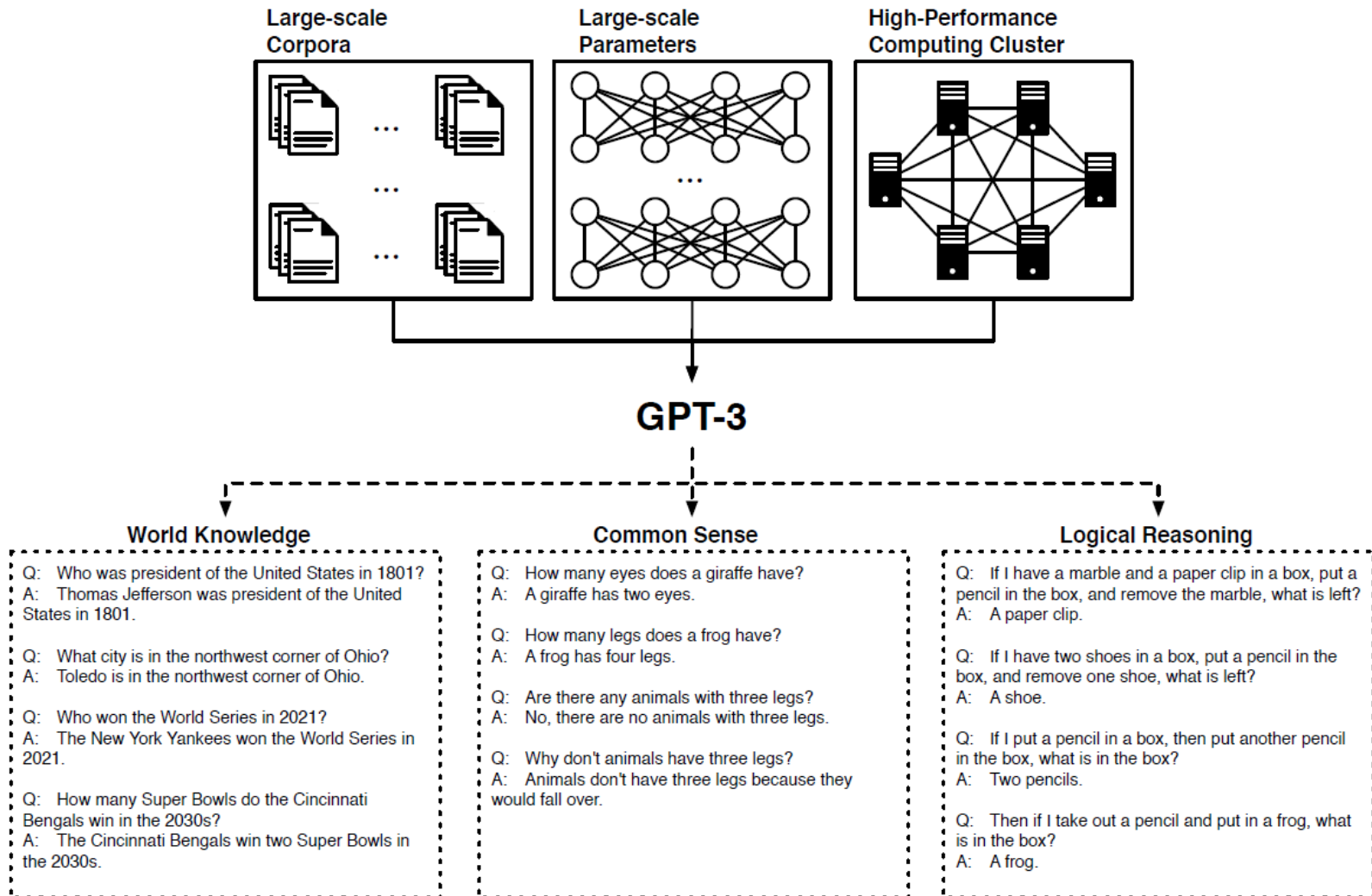


(a) Evaluation on language understanding benchmark GLUE.



(b) Manual evaluation on dialogue systems.

GPT-3: prompt engineering



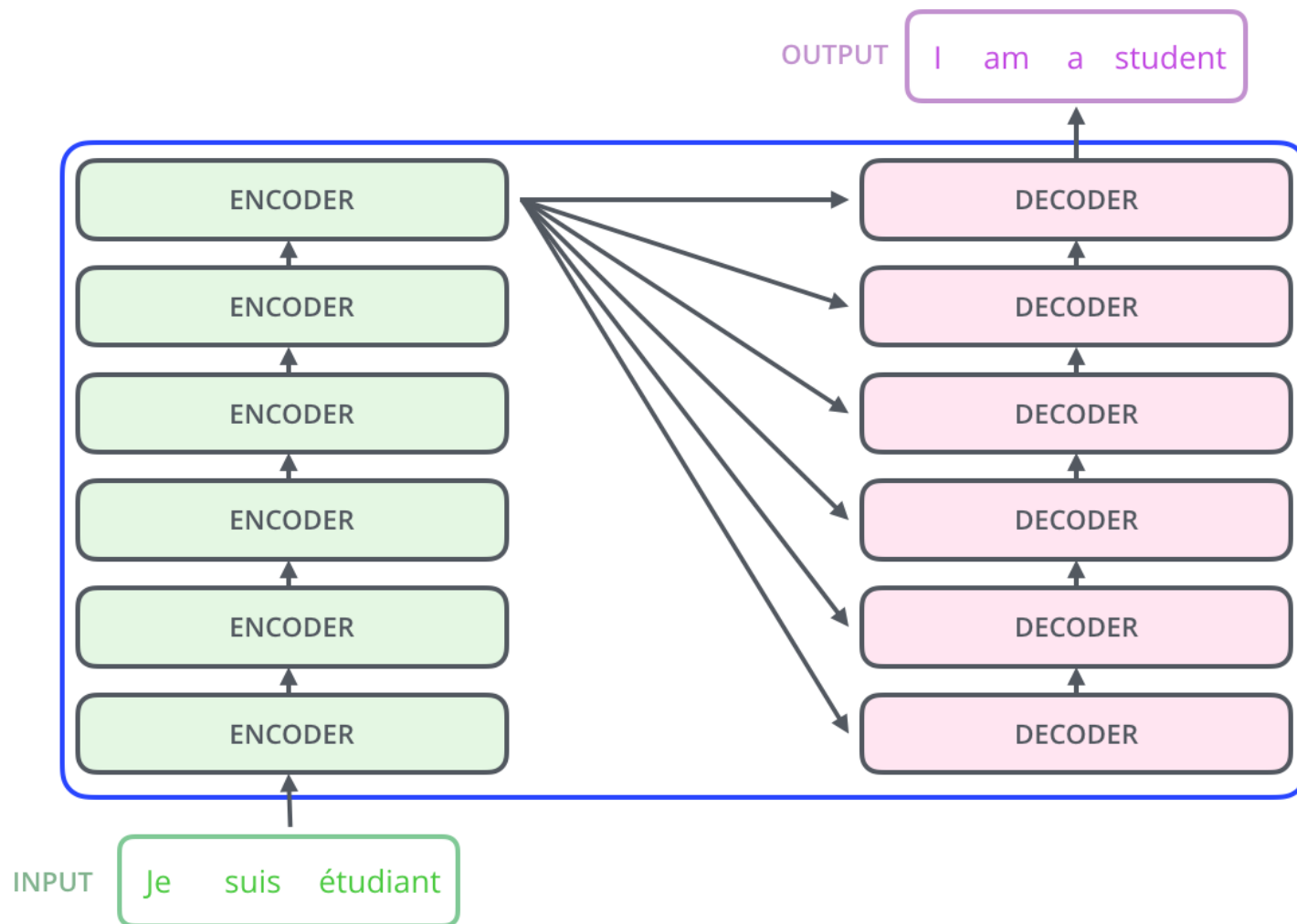
План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

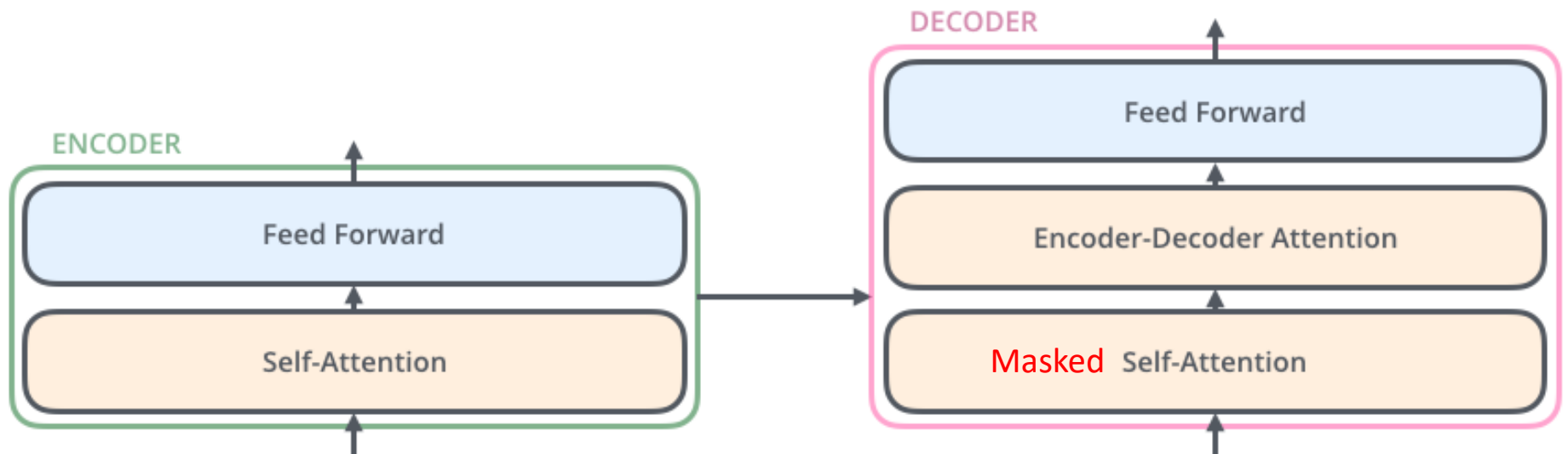
GPT

- GPT = Generative Pre-trained Transformer (или Generative Pre-Training)
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever (OpenAI). Improving Language Understanding by Generative Pre-Training. 2018
 - Блог: <https://openai.com/blog/language-unsupervised> (11.06.2018)
 - Статья: [ссылка](#)
 - Исходный код: <https://github.com/openai/finetune-transformer-lm>
- Архитектура – 12 блоков декодировщика (decoder) Transformer
 - Vaswani et al. (Google Brain) – [Attention Is All You Need](#) // NIPS, 2017
 - The Illustrated Transformer (by Jay Alammar):
 - Оригинал: <https://jalammar.github.io/illustrated-transformer/>
 - Перевод: <https://habr.com/ru/post/486358/>

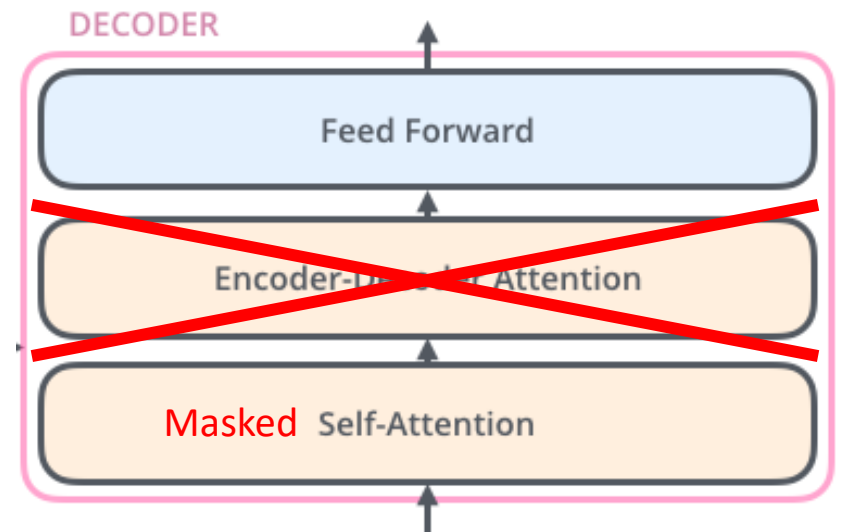
Transformer – архитектура



Transformer – один блок



GPT



GPT

- Unsupervised pre-training:

$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}) \rightarrow \max,$$

где $U = \{u_1, \dots, u_n\}$ – корпус токенов

- Supervised fine-tuning:

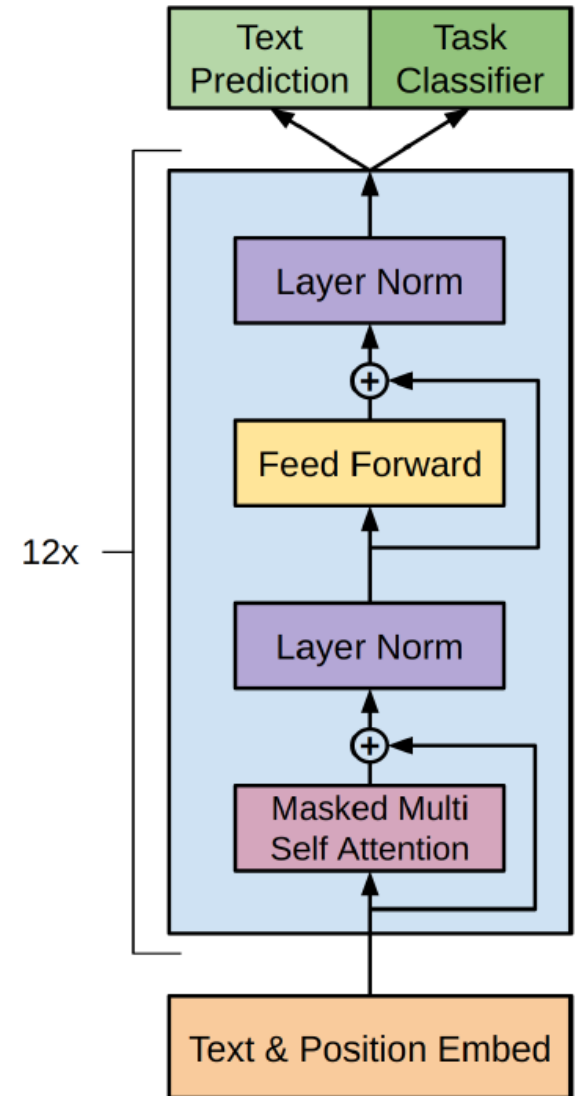
$$L_2(C) = \sum_{(x,y)} \log P(y | x_1, \dots, x_m) \rightarrow \max,$$

где C – размеченный датасет, каждый пример которого включает:

$\{x_1, \dots, x_m\}$ – последовательность входных токенов,
 y – метка входных токенов

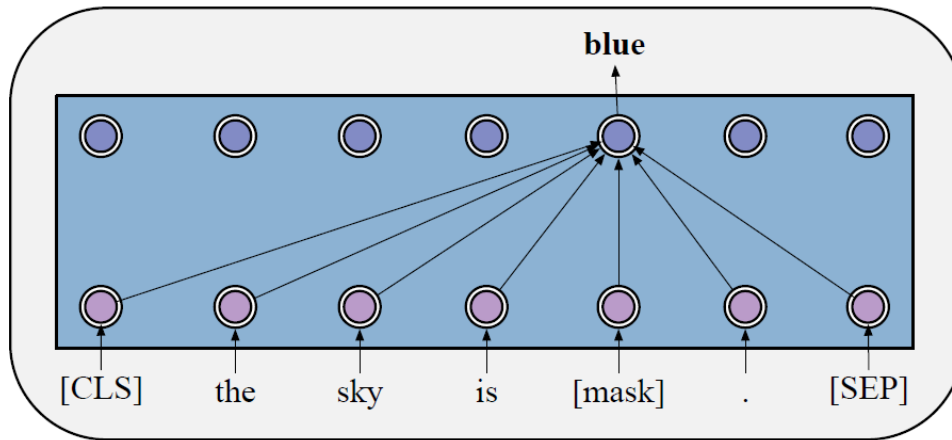
- Итоговая целевая функция (на этапе fine-tuning):

$$L_3(C) = L_2(C) + \lambda L_1(C)$$

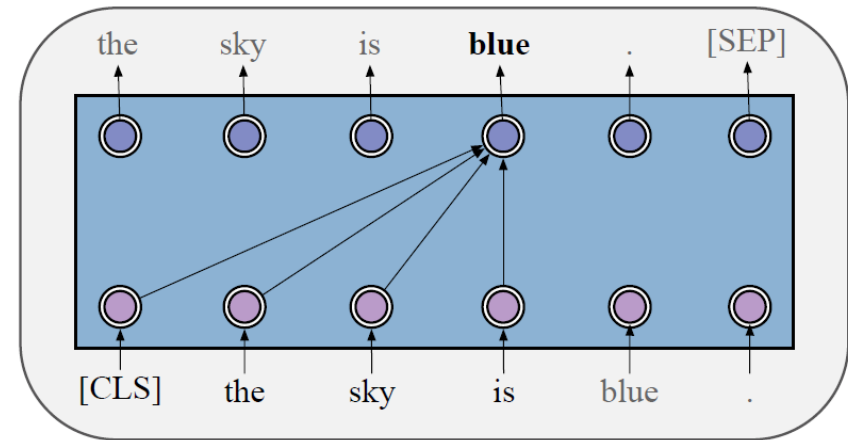


BERT vs. GPT

- BERT – двунаправленная модель
 - задачи Masked Language Modelling и Next Sentence Prediction
- GPT – модель, учитывающая предыдущий контекст (left-to-right)
 - задача Next Word Prediction (Causal Language Modelling)



BERT



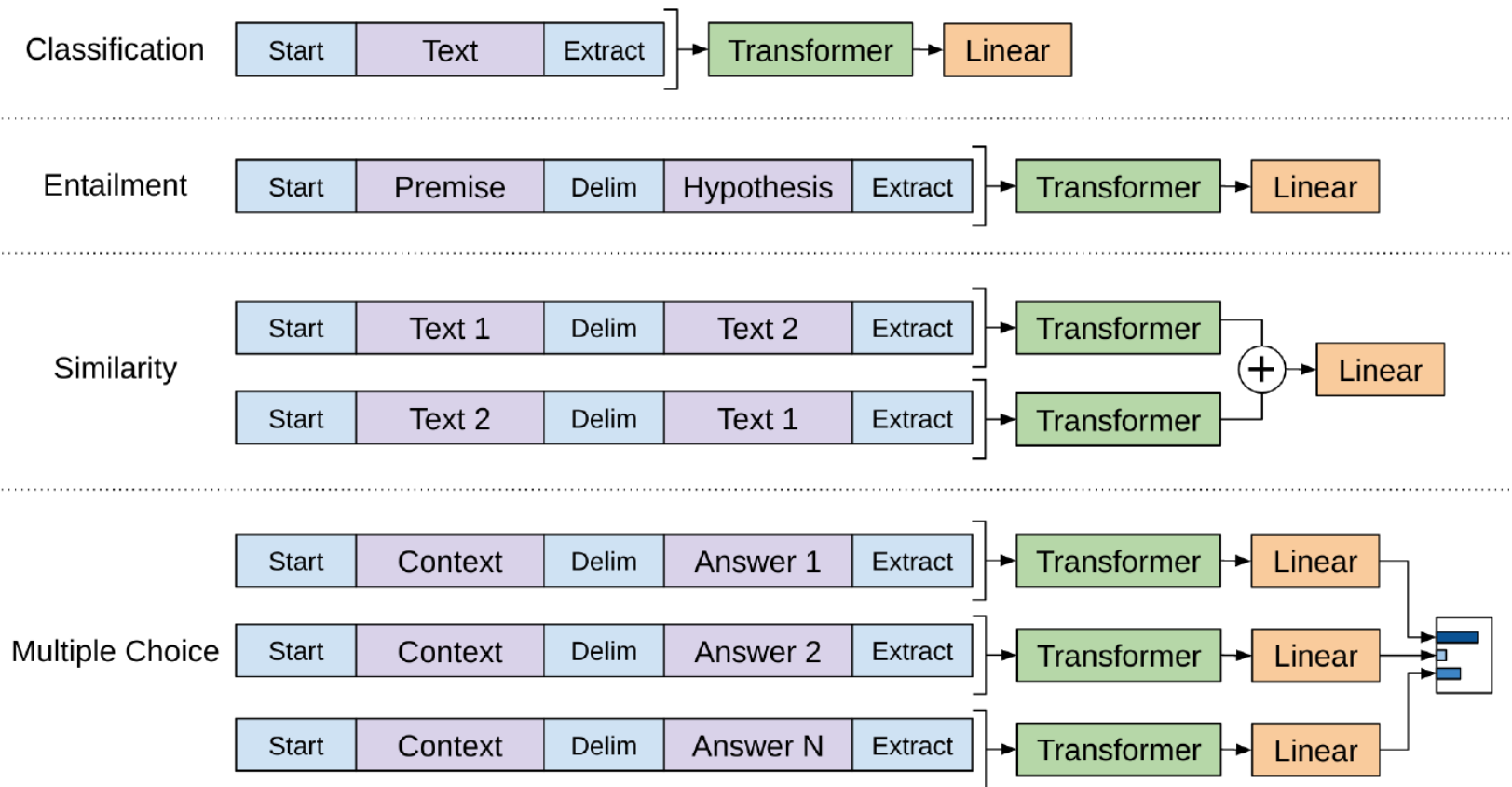
GPT

GPT – обучение

- BooksCorpus – 7000 книг разных жанров (5 ГБ)
- 117 миллионов параметров (117M)
- 30 дней, 8 P600
- Словарь: Byte Pair Encoding (BPE), 40,000 токенов
 - В BERT – WordPiece, но разницы большой нет
- Метод оптимизации: Adam
- Unsupervised pre-training:
 - Скорость обучения (максимальная): $2,5 \cdot 10^{-4}$
 - Количество эпох: 100
 - Размер батча: 64
- Supervised fine-tuning:
 - Скорость обучения (максимальная): $6,25 \cdot 10^{-5}$
 - Количество эпох: 3
 - Размер батча: 32

GPT – задачи

- Для 9 задач (корпусов) из 12 получены State-of-the-Art результаты



План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

GPT-2

- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever (Open AI). Language Models are Unsupervised Multitask Learners. 2019
 - Блог: <https://openai.com/blog/better-language-models> (14.02.2019)
 - Статья: [ссылка](#)
 - Исходный код: <https://github.com/openai/gpt-2>
 - Демо Hugging Face: <https://transformer.huggingface.co/doc/gpt2-large>
 - Демо AllenNLP: <https://demo.allennlp.org/next-token-lm>
- Только 05.11.2019 выложили полную модель (1.5B = 1.5 млрд)
 - <https://www.openai.com/blog/gpt-2-1-5b-release>

Режимы обучения (1)

- Fine-tuning:



Режимы обучения (2)

- Few-shot learning:

The diagram illustrates a few-shot learning prompt structure. It consists of five numbered lines within a light blue rounded rectangle. Line 1 is the task description. Lines 2, 3, and 4 are examples. Line 5 is the prompt. Arrows on the right point from labels to the corresponding lines: 'task description' to line 1, 'examples' to lines 2-4, and 'prompt' to line 5.

Line	Text	Label
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

Режимы обучения (3)

- One-shot learning:

1 Translate English to French:

← *task description*

2 sea otter => loutre de mer

← *example*

3 cheese =>

← *prompt*

.....

Режимы обучения (4)

- Zero-shot learning:

1 Translate English to French:

← *task description*

2 cheese =>

← *prompt*

.....

GPT-2

- Unsupervised multitask learning
- Zero-shot task transfer
- Обучающий корпус (WebText):
 - Ссылки с Reddit (проверка пользователями)
 - Нет статей Википедии
(чтобы исключить пересечение с тестовыми корпусами)
 - Более 8 млн документов (40 ГБ)
 - 10 МБ французских текстов
- 256 cloud TPUs
 - Время, детали и стоимость обучения не раскрывается

GPT-2 vs. GPT

- Byte Pair Encoding (BPE) над байтами, не над символами
- Словарь: 50 257 токенов (вместо 40 000)
- Размер входа: 1024 токена (вместо 512)
- Размер батча: 512 (вместо 64)
- Слои нормализации перемещены на вход каждого блока
- Дополнительный слой нормализации добавлен после последнего блока внутреннего внимания
- Количество параметров увеличено (4 варианта модели)

GPT-2: варианты моделей

Количество параметров	Количество слоев	Размерность скрытых состояний	Примечание
117M	12	768	Аналог GPT
345M	24	1024	Аналог BERT _{LARGE}
762M	36	1280	
1542M	48	1600	GPT-2

GPT-2: результаты (zero-shot)

- **Language Modeling:** для 7 из 8 датасетов улучшены State-of-the-Art оценки
- **Reading Comprehension:** соответствует baselines (55 F1), значительно ниже SOTA (BERT – 89 F1)
- **Summarization:** добавляли **TL;DR:** после текста
 - На уровне трех случайных предложений
- **Translation:** французско-английский перевод
 - На уровне простых unsupervised baselines
- **Question Answering:** на уровне простых baselines

План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

GPT-3

- Tom Brown, Alec Radford, Ilya Sutskever et al. (31 соавтор). Language Models are Few-Shot Learners. Май 2020
 - Версия статьи на NeurIPS (25 стр.): [ссылка](#) (декабрь 2020)
 - Версия статьи на arXiv (75 стр.): [ссылка](#) (июль 2020)
 - OpenAI API: <https://openai.com/blog/openai-api>
 - 6 центов за 1000 токенов (4 рубля 25 коп.)
 - Данные: <https://github.com/openai/gpt-3>

GPT-3

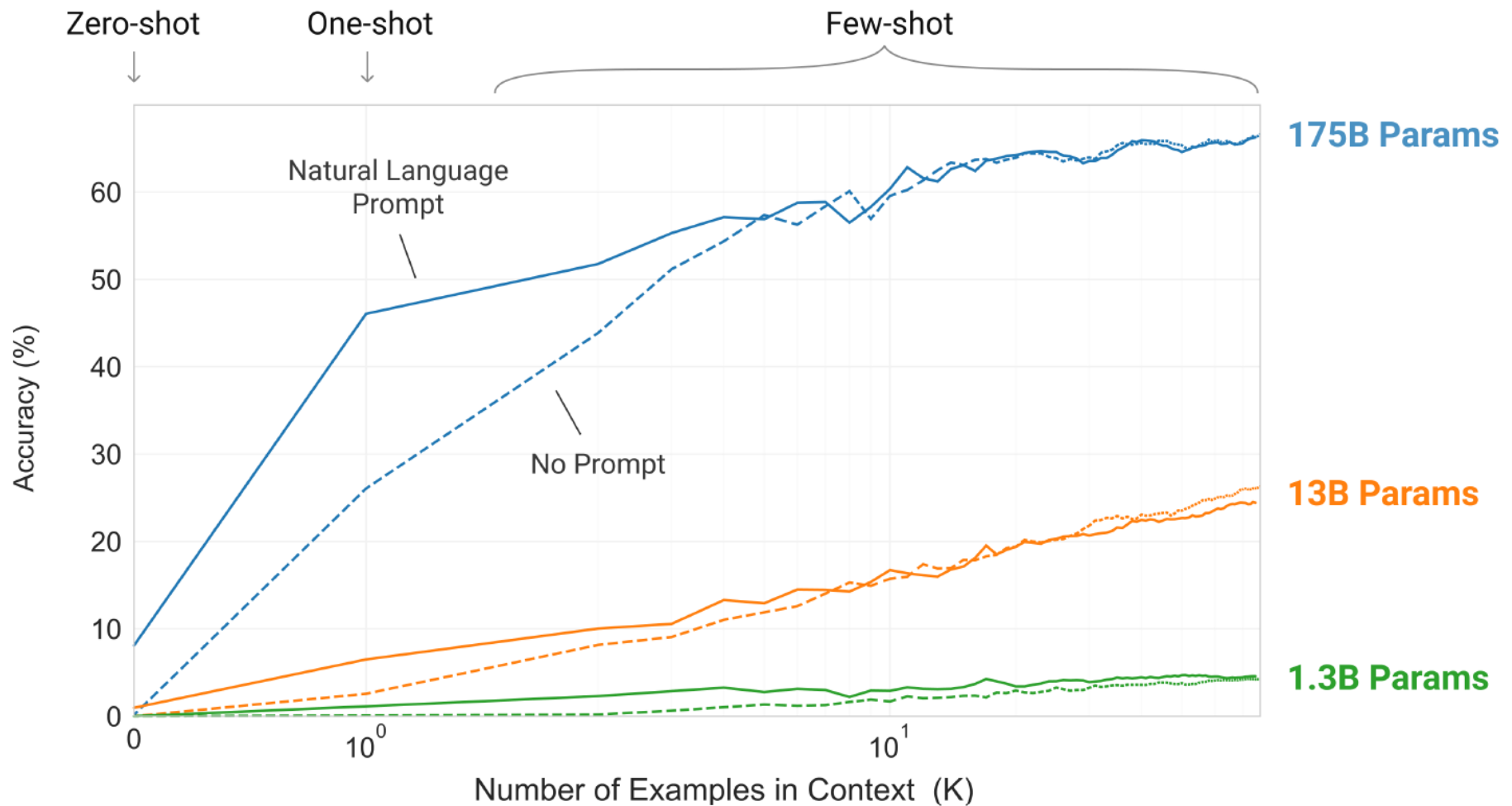
- Архитектура – как в GPT-2, кроме:
 - размер входа: 2048 токенов
 - перемежающиеся блоки с плотным (dense) и разреженным (sparse) вниманием
- Данные для обучения: 500 млрд токенов, 560 ГБ
 - Common Crawl, WebText, Books, English Wikipedia
- Статистика по языкам (по словам) ([ссылка](#)):
 - Английский – 92.7%
 - Французский – 1.8%
 - Русский – 0.19%

GPT-3

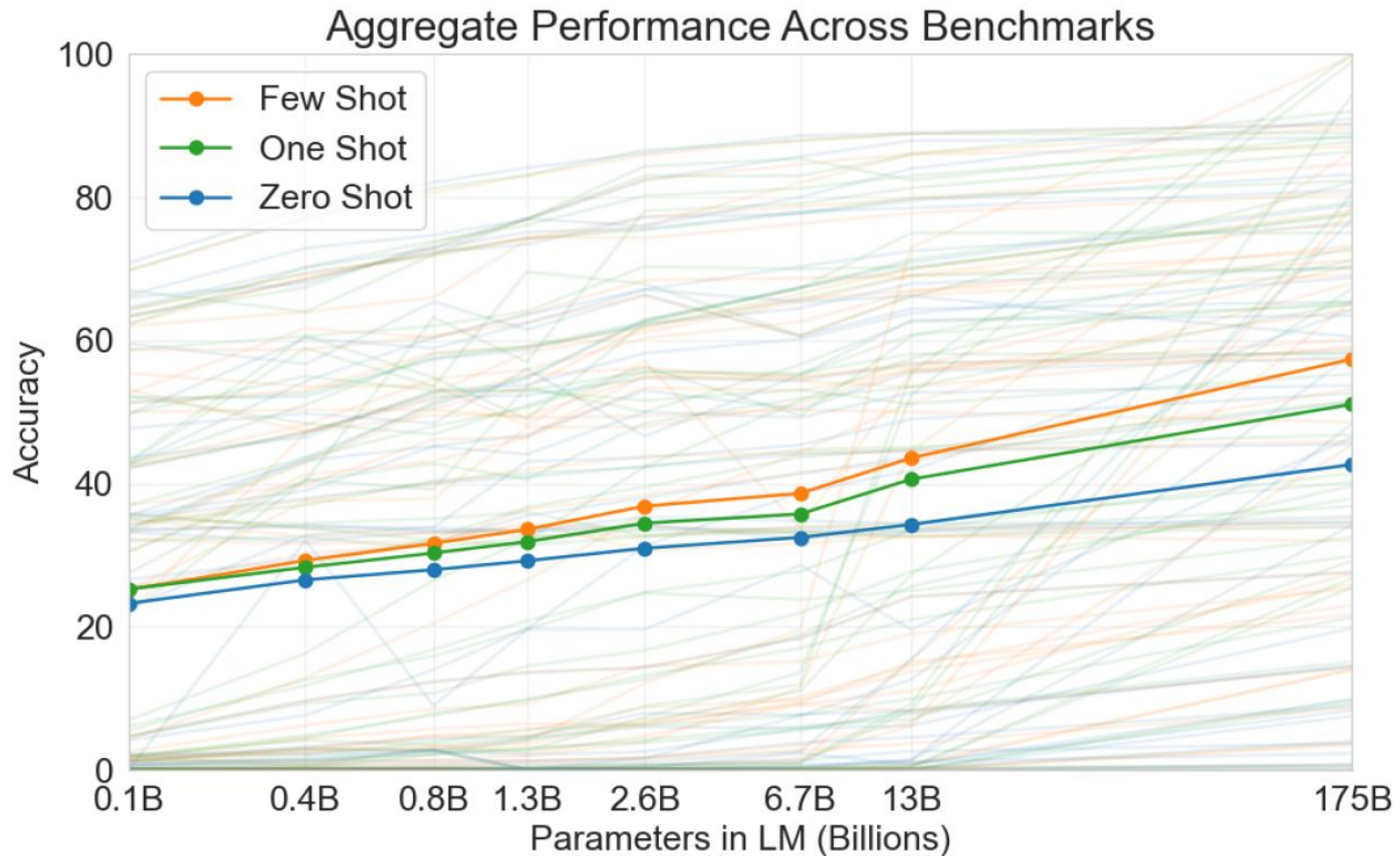
- Обучение: кластер V100
- Стоимость обучения: 4.6 ... 12 млн \$
- Чем больше модель, тем больше размер батча и меньше скорость обучения

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

GPT-3



GPT-3: 42 datasets



GPT-3: результаты (1)

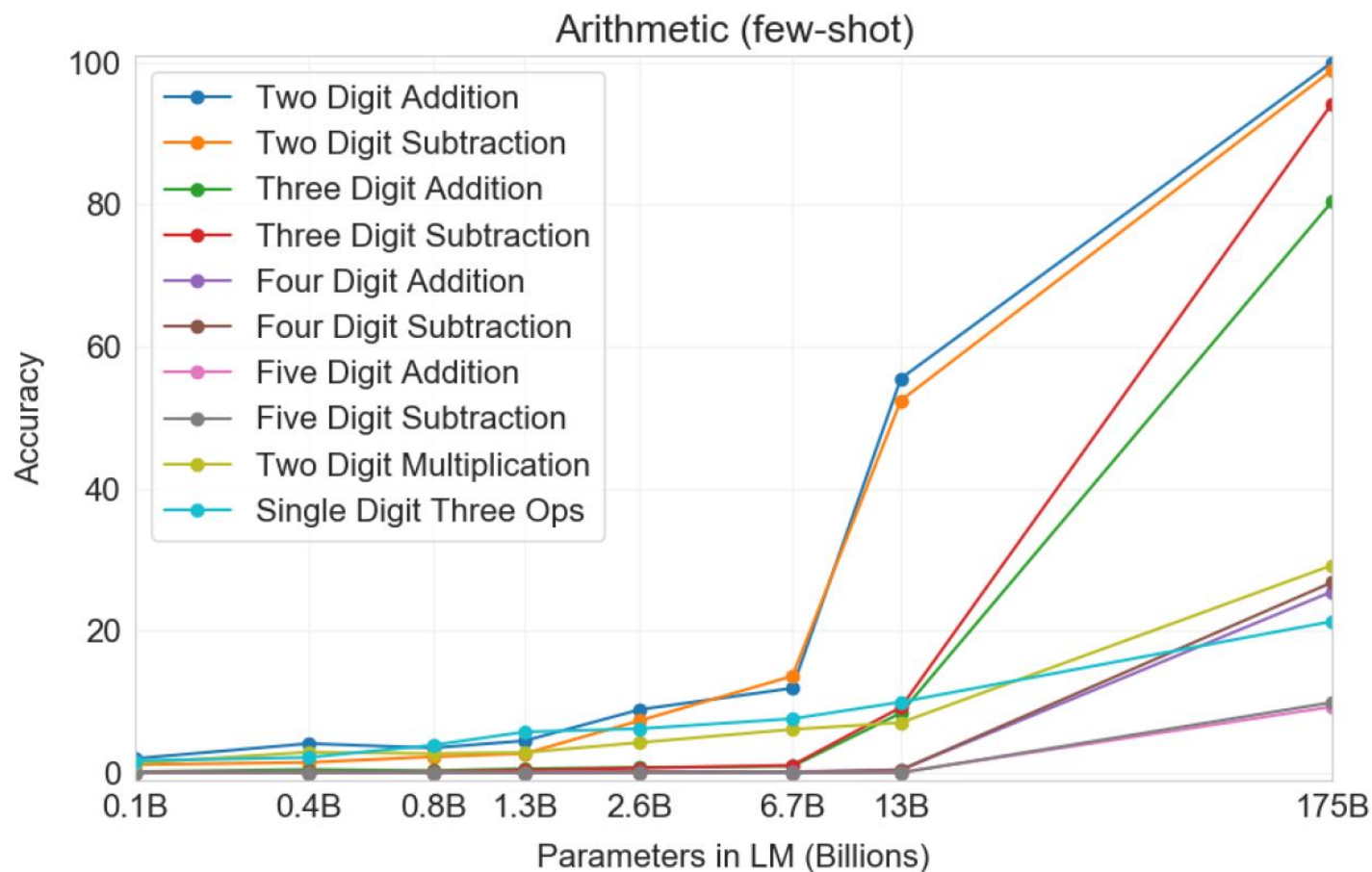
- Language Modeling, Cloze, and Completion Tasks – 2 из 4
- Closed Book Question Answering – 1 из 3
- Translation – 3 из 6 (unsupervised SOTA)
- Winograd-Style Tasks (корелация) – близко к SOTA
- Common Sense Reasoning – 1 из 4
- Reading Comprehension – 0 из 6
- SuperGLUE – хуже SOTA (71.8 vs. 89.0) ([ссылка](#)),
но лучше fine-tuned BERT_{LARGE}
- Natural Language Inference (NLI) – почти как случайное
угадывание

GPT-3: результаты (2)

Synthetic and Qualitative Tasks:

- Arithmetic
- Word Scrambling and Manipulation Tasks
- SAT Analogies
- News Article Generation
- Learning and Using Novel Words
- Correcting English Grammar

GPT-3: арифметика



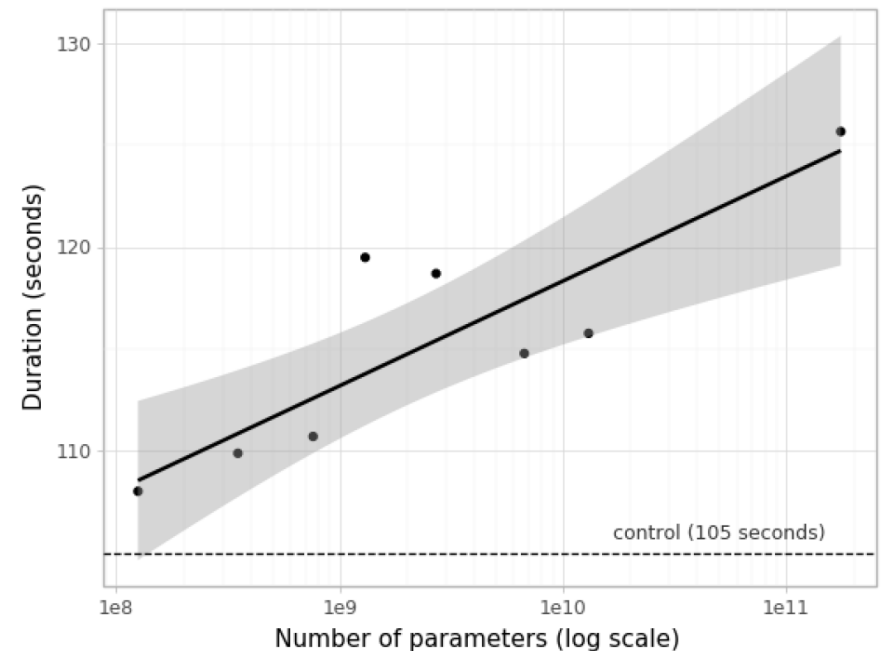
GPT-3: генерация статей (1)

- Эксперимент: сможет ли человек отличить, сгенерирована статья или написана человеком?
- Взяли 25 статей с сайта <https://www.newser.com>
 - Средняя длина заголовков и подзаголовков – 215 слов
- Если только одно предложение в качестве подсказки – GPT-3 воспринимает его как твит и генерирует ответ
 - Поэтому предоставляли три статьи в качестве контекста
- Сгенерировали продолжения для 25 статей
 - Результат отформатировали
 - Средняя длина текста – 200 слов
- *Control* – модель 160М без контекста
- Аннотаторы – 80 американцев
- Случайное угадывание: Accuracy = 50%

GPT-3: генерация статей (2)

	Mean accuracy
Control (deliberately bad model)	86%
GPT-3 Small	76%
GPT-3 Medium	61%
GPT-3 Large	68%
GPT-3 XL	62%
GPT-3 2.7B	62%
GPT-3 6.7B	60%
GPT-3 13B	55%
GPT-3 175B	52%

Average time spent trying to detect model generated news article



GPT-3: демо

- Simon O'Regan – GPT-3: Demos, Use-cases, Implications
 - [Ссылка](#)
- Awesome GPT-3
 - [Ссылка](#)
- 21 OpenAI GPT-3 Demos and Examples to Convince You that AI Threat is Real, or is it ?
 - [Ссылка](#)
- Demos of OpenAI's GPT-3 – GPT Crush
 - [Ссылка](#)

GPT-3: ограничения

- Проблемы при генерации текста:
 - фактические ошибки («галлюцинации»)
 - повторы
 - логические нестыковки
 - нетипичные фразы
- Проблемы с «физикой здравого смысла»:
 - «If I put cheese into the fridge, will it melt?»
- Плохо справляется с задачами, где нужна двунаправленность
 - Логическая связь предложений
 - Ответы на вопросы по пониманию текста
- Fine-tuning больше подходит для двунаправленных моделей
- Недостаток информации о реальном мире (видео, физические действия)
- Большой объем информации во время предобучения (в отличие от человека)
- Плохая интерпретируемость результатов
- Предвзятость, основанная на обучающих данных (пол, раса, религия, ...)

GPT-3: аналоги

- Английский:
 - GPT-J-6B – 6 млрд параметров ([ссылка](#)) – доступно
 - GPT-NeoX-20B – 20 млрд параметров ([ссылка](#)) – доступно
 - Wu Dao 2.0 – 1.75 трлн, мультимодальная ([Хабр](#)) – недоступно
- Русский:
 - ruGPT3 (Сбер) (<https://github.com/ai-forever/ru-gpts>) – доступно
 - YaLM (Яндекс) ([Хабр1](#), [Хабр2](#), [Балабоба](#)) – доступно
- Китайский:
 - PanGu (<https://arxiv.org/abs/2104.12369>)
- Французский:
 - PAGnol (<https://lair.lighton.ai/pagnol>)
- Корейский:
 - HyperCLOVA (<https://github.com/clovaai>)

План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

ruGPT3: модели

- Модели:
 - ruGPT3XL – 1.3B (разреженное внимание, sparse attention)
 - ruGPT3Large – 760M (размер входа 2 048 токенов)
 - ruGPT3Medium – 350M (размер входа 2 048 токенов)
 - ruGPT3Small – 125M (размер входа 2 048 токенов)
 - ruGPT2Large – 774M (размер входа 1 024 токенов)
- Ссылки:
 - <https://github.com/ai-forever/ru-gpts>
 - <https://sbercloud.ru/ru/warp/gpt-3>
 - <https://habr.com/ru/company/sberbank/blog/524522/>
 - <https://huggingface.co/models?search=ai-forever>
 - <https://russiannlp.github.io/rugpt-demo/>
- Russian SuperGLUE ([ссылка](#))

ruGPT3: обучение

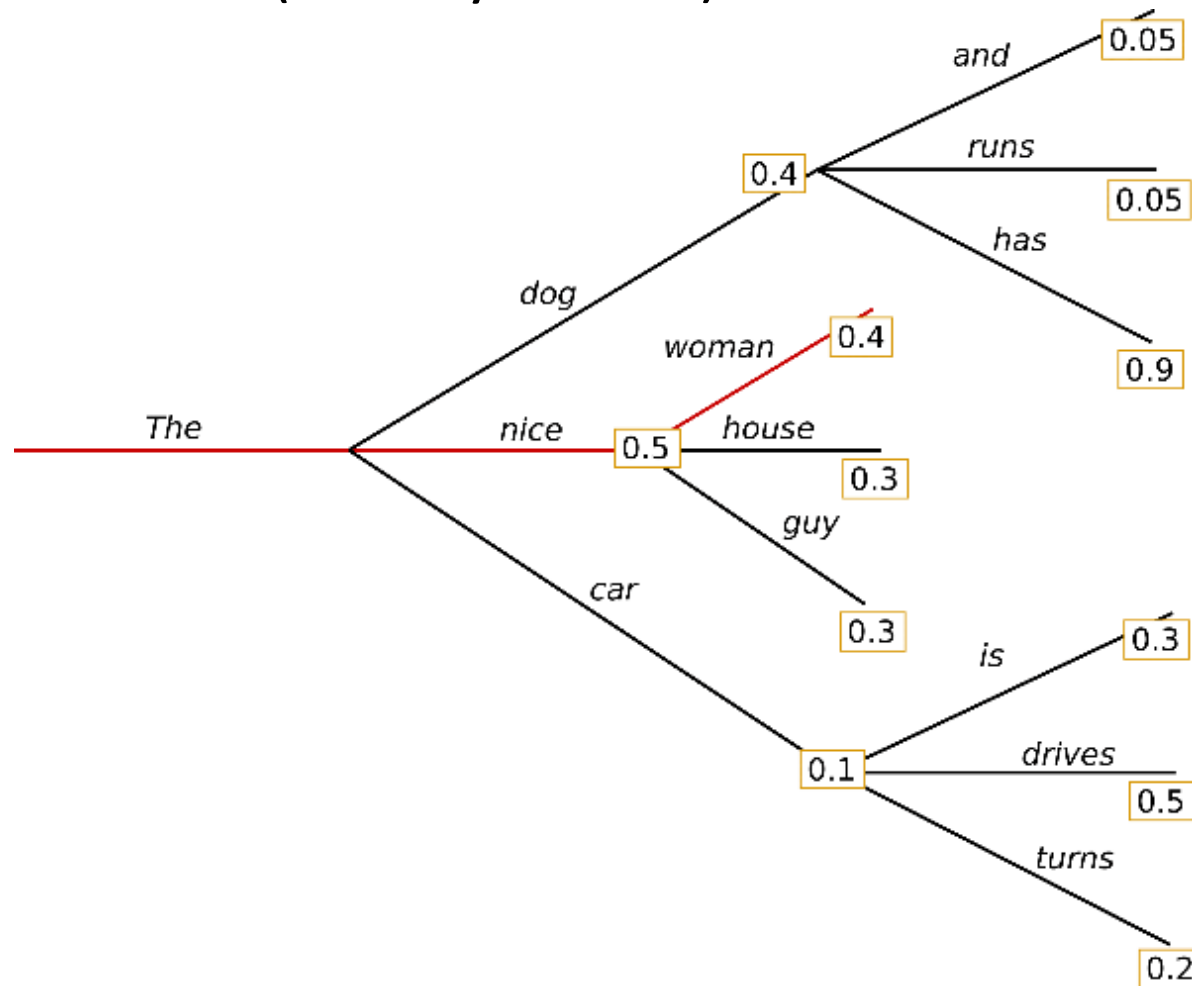
- Обучающий корпус:
 - 600 ГБ – Википедия (ru, en), художественная литература, новости, вопросы-ответы, Pikabu, научно-популярные статьи, банки, github и StackOverflow
 - Соотношение русского и других языков: 9 к 1
- Обучение:
 - Суперкомпьютер Кристофари ([ссылка](#))
 - 75 вычислительных узлов DGX-2 (в каждом по 16 карт NVidia V100)
 - использовался не полностью
 - Megatron (NVIDIA, [ссылка](#))
 - DeepSpeed (Microsoft, [ссылка](#))

План

- Языковые модели
- GPT
- GPT-2
- GPT-3
- ruGPT3
- Стратегии генерации текста

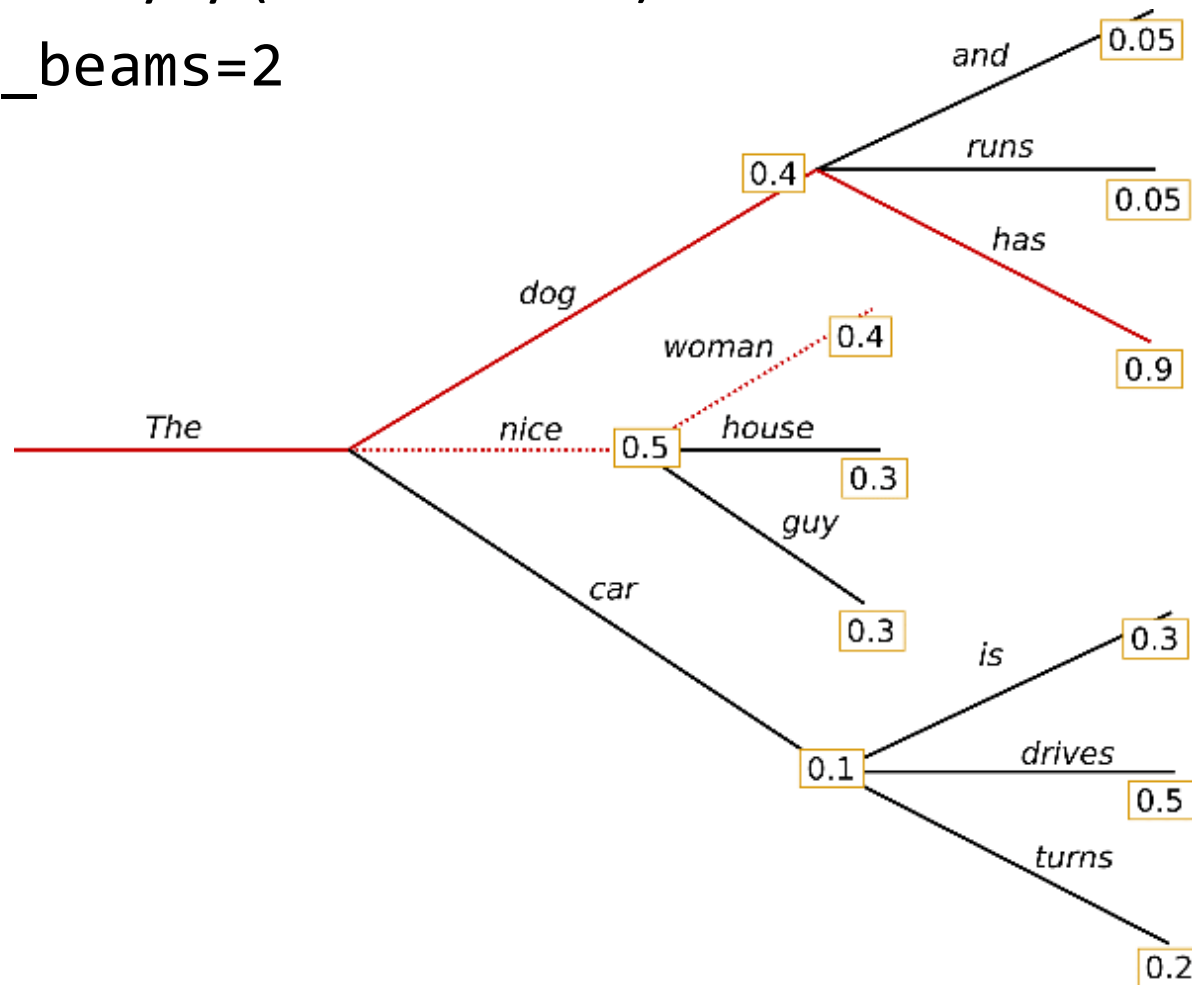
Стратегии генерации текста

- Жадный поиск (Greedy Search)



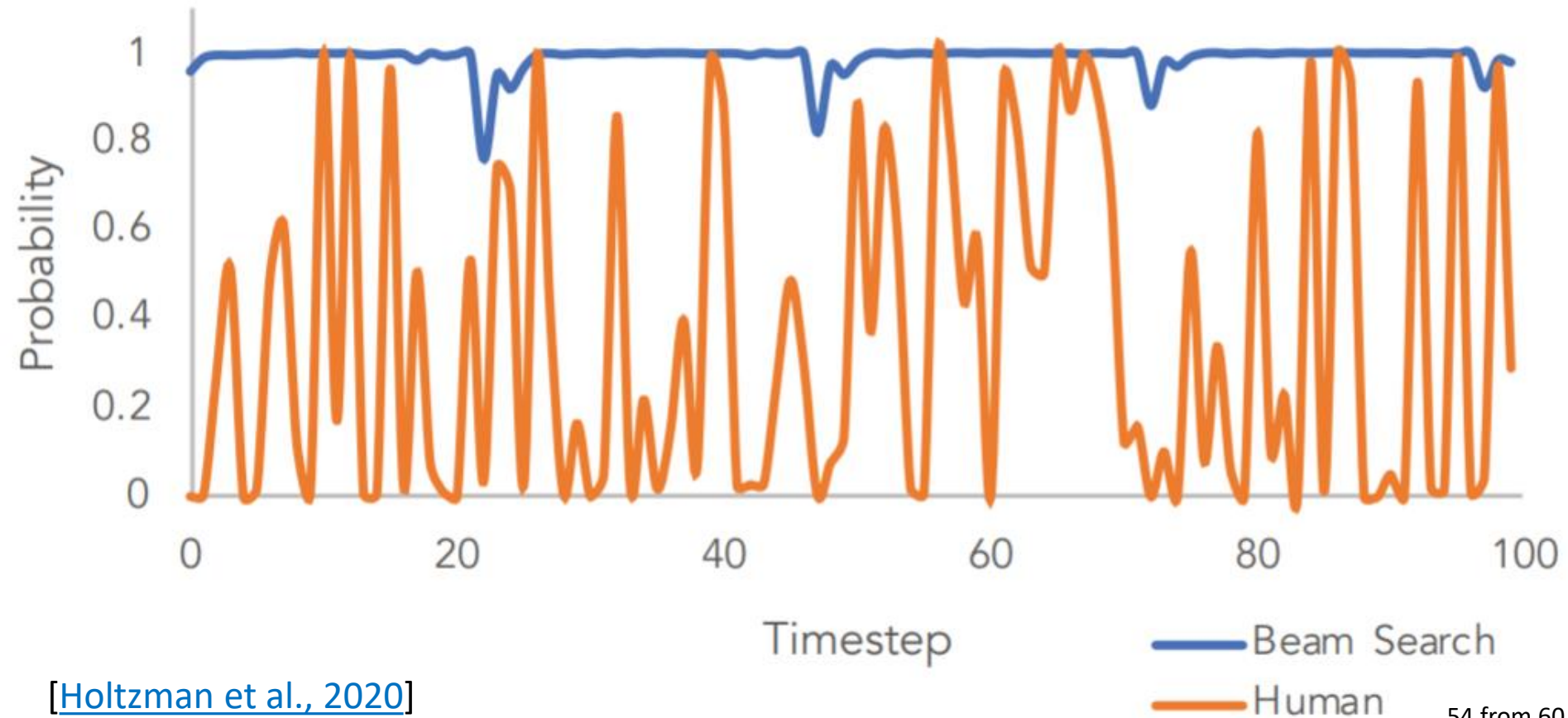
Стратегии генерации текста

- Поиск по лучу (Beam Search)
 - num_beams=2



Стратегии генерации текста

Beam Search Text is Less Surprising



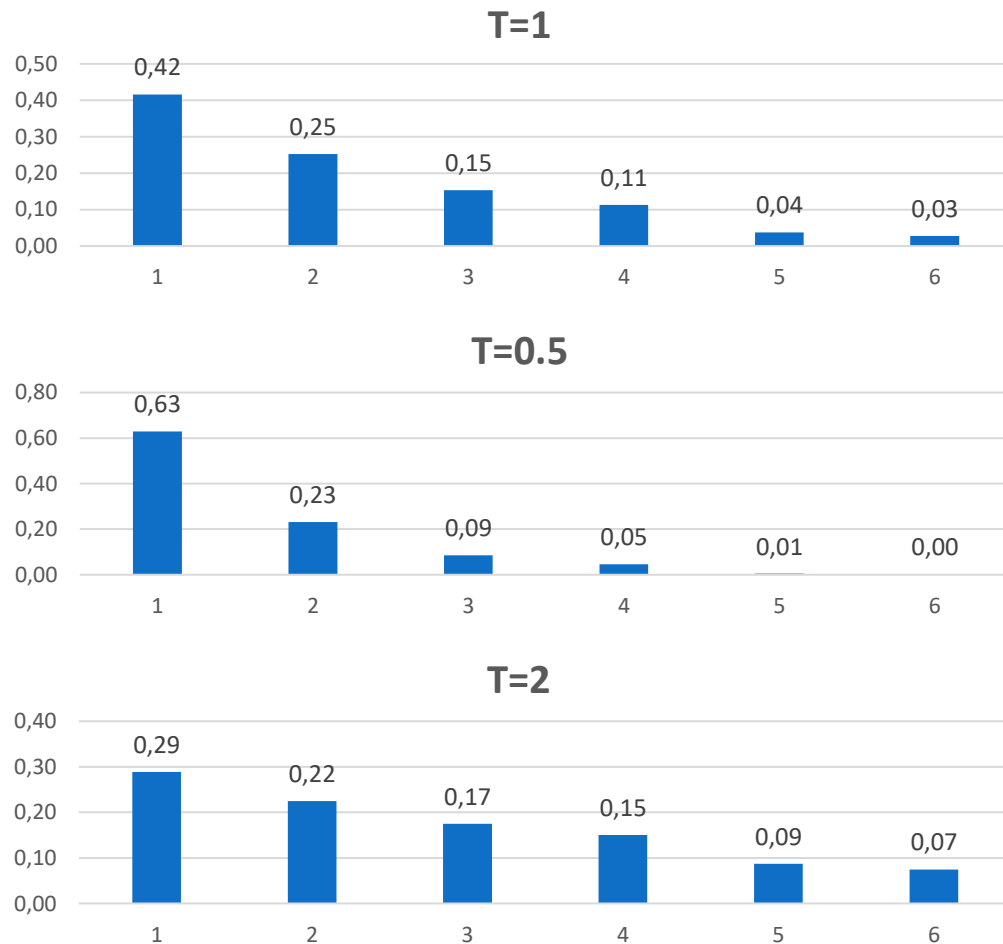
Стратегии генерации текста

- Софтмакс:

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

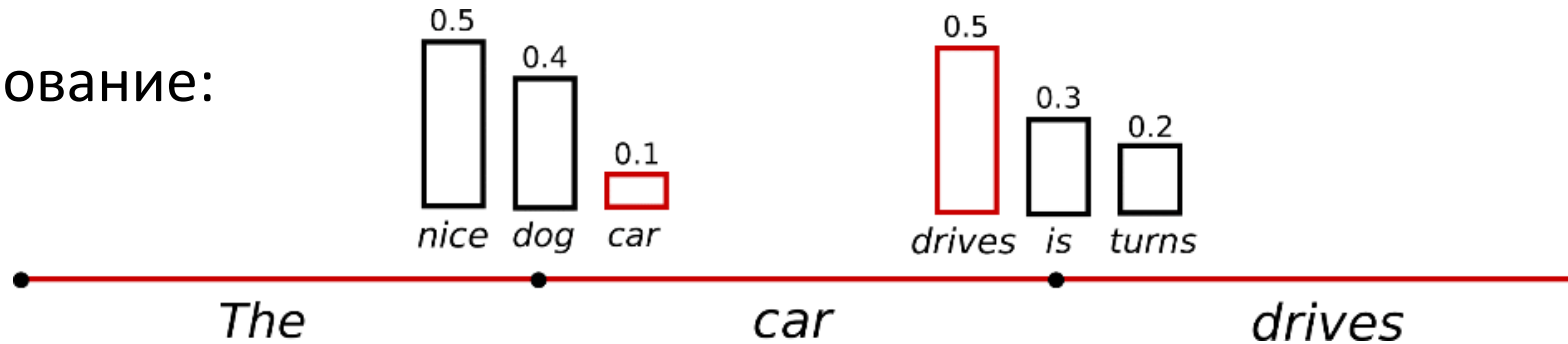
- Семплирование с температурой:

$$p_i = \frac{e^{\frac{z_i}{T}}}{\sum_j e^{\frac{z_j}{T}}}$$

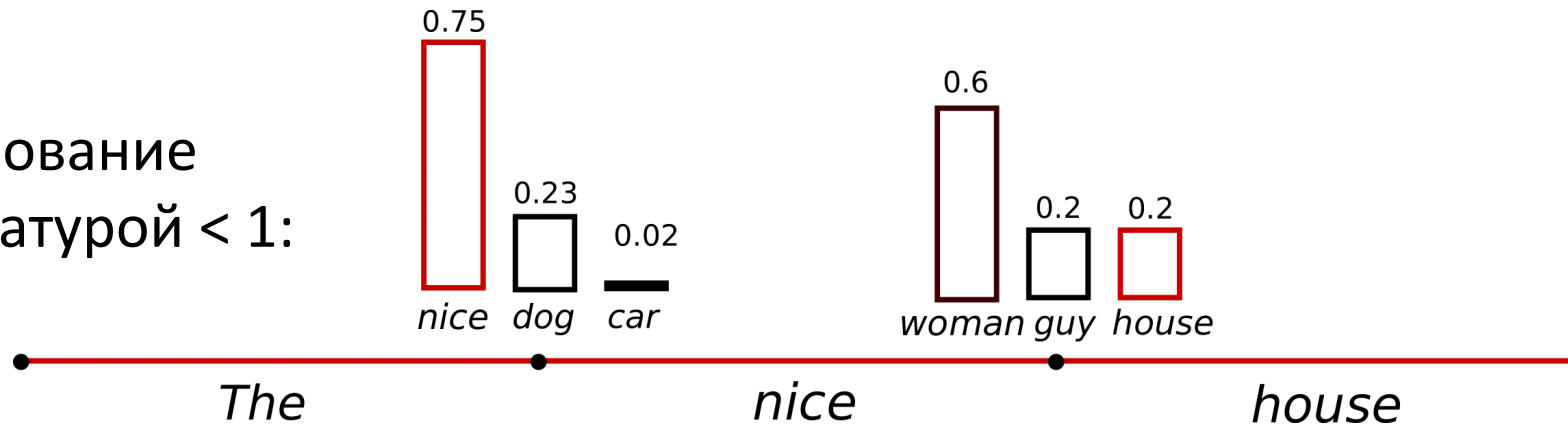


Стратегии генерации текста

Семплирование:

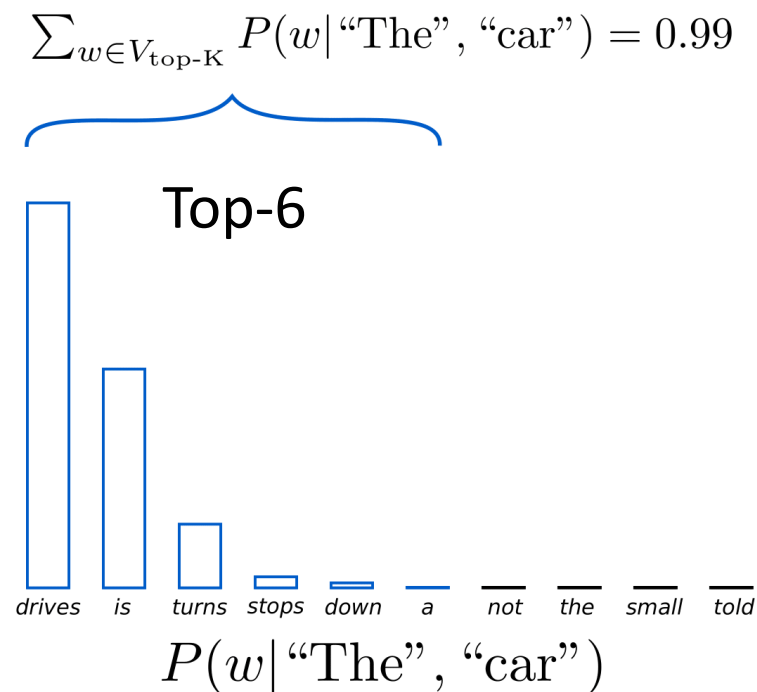
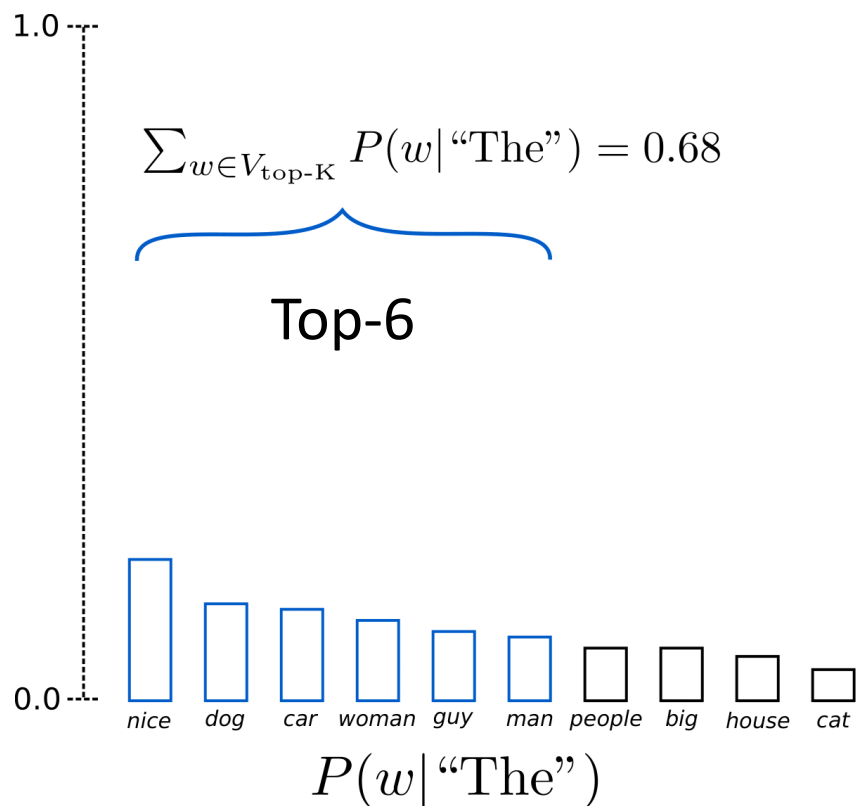


Семплирование
с температурой < 1 :



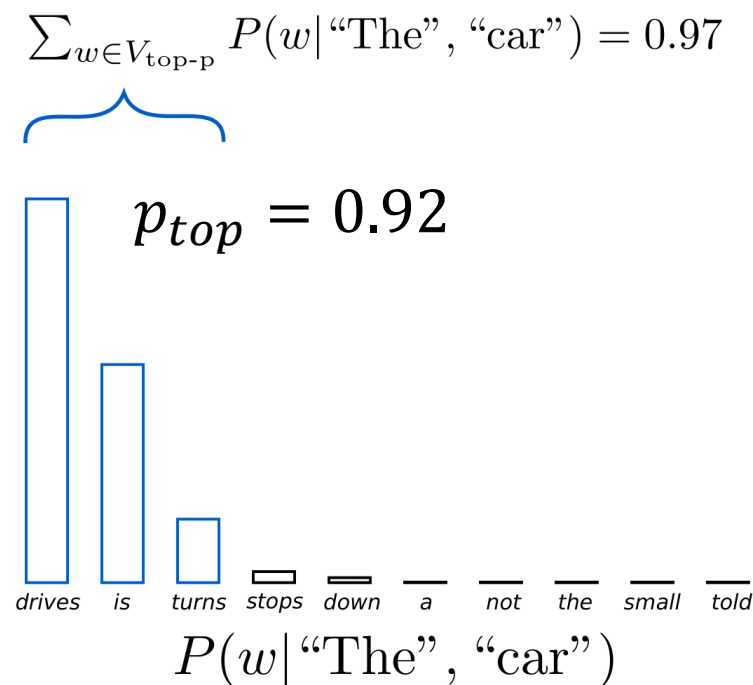
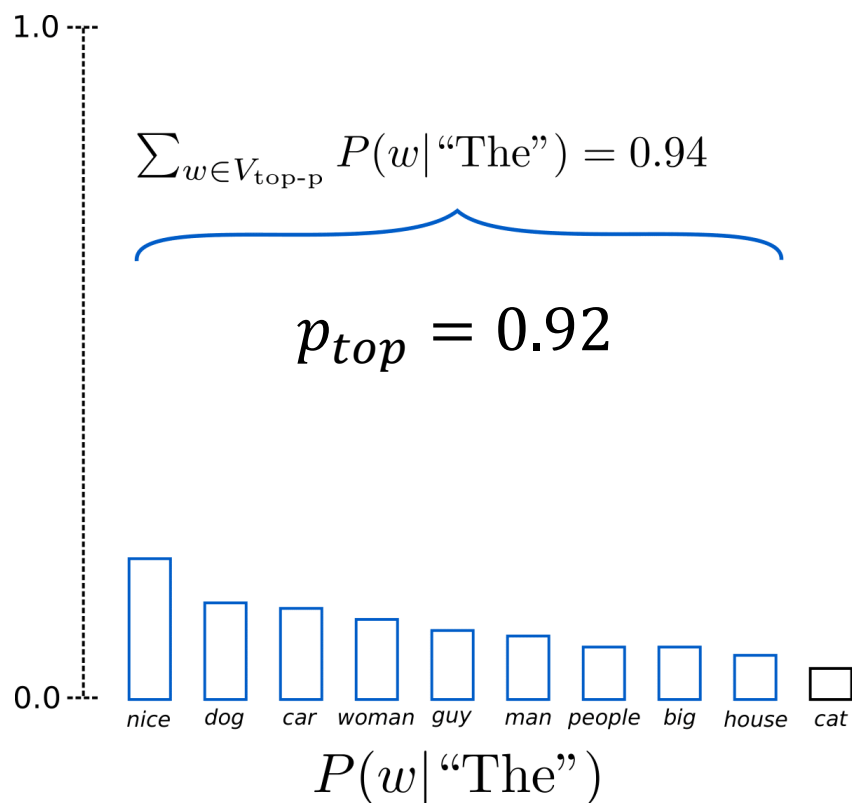
Стратегии генерации текста

- Top-K семплирование [[Fan et al., 2018](#)]

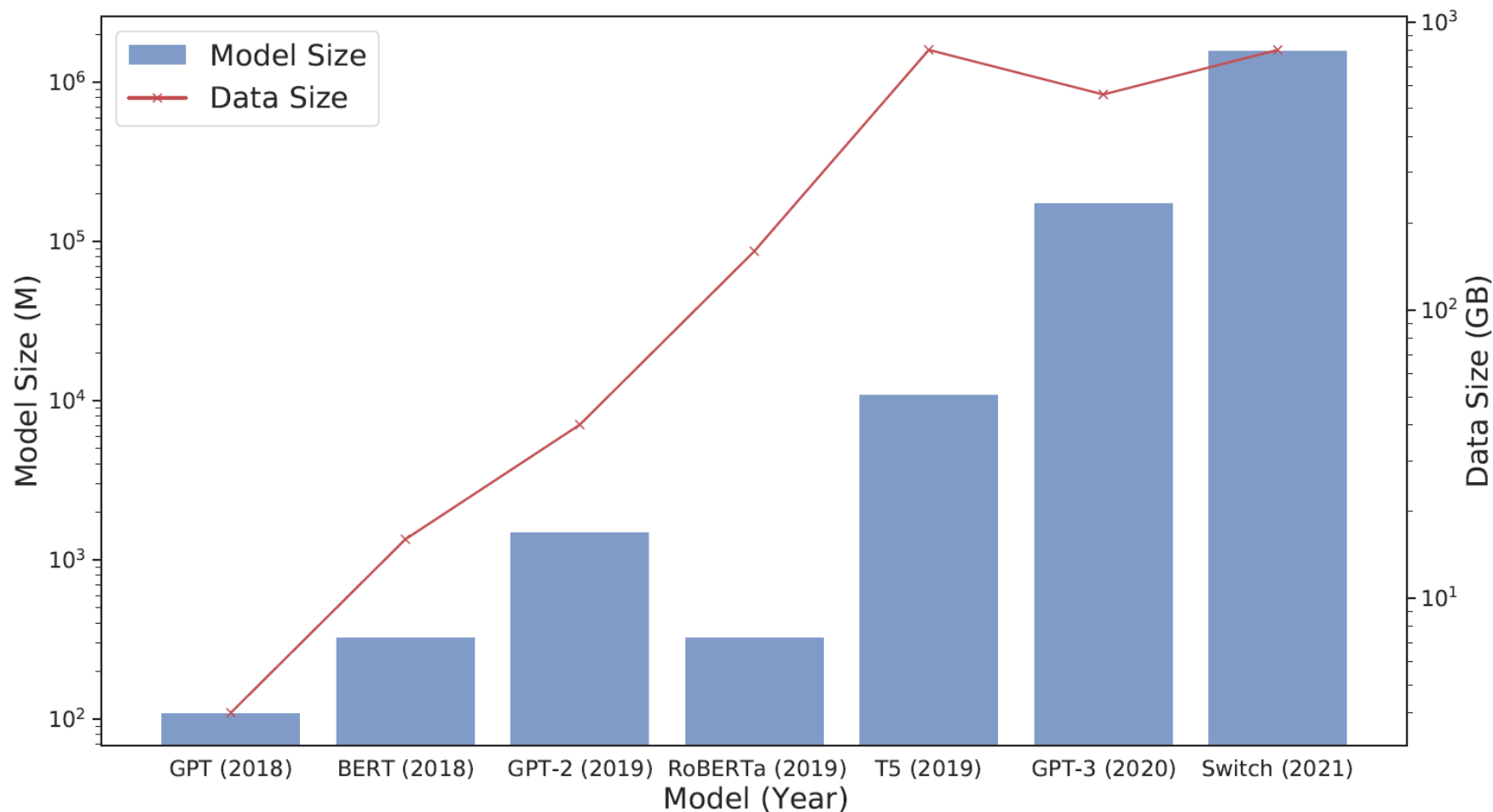


Стратегии генерации текста

- Top-p (nucleus) семплирование



Размеры моделей и датасетов



Полезные ссылки

rugpt3.ipynb

- NLP Course (by Lena Voita)
 - https://lena-voita.github.io/nlp_course.html
- Summary of the models (by Hugging Face)
 - https://huggingface.co/transformers/model_summary.html
- Must-Read Papers on Pre-trained Language Models (PLMs)
 - <https://github.com/thunlp/PLMpapers>
- The Illustrated GPT-2 (by Jay Alammar)
 - Оригинал: <https://jalammar.github.io/illustrated-gpt2>
 - Перевод: <https://habr.com/ru/post/490842>
- Токенизация на подслова (Александр Дьяконов)
 - <https://dyakonov.org/2019/11/29/токенизация-на-подслова-subword-tokenization>
- How to generate text using different decoding methods
 - <https://huggingface.co/blog/how-to-generate>

ChatGPT

План

- ChatGPT
 - Примеры
 - Последствия
- Диалоговые модели
 - OpenAI InstructGPT
 - DeepMind Sparrow
- Заключение
 - Распознавание сгенерированных текстов
 - Ограничения ChatGPT
 - Выводы



Andrej Karpathy ✓

@karpathy

The hottest new programming language is English

[Перевести твит](#)

11:14 PM · 24 янв. 2023 г. · **1,9 млн** просмотров

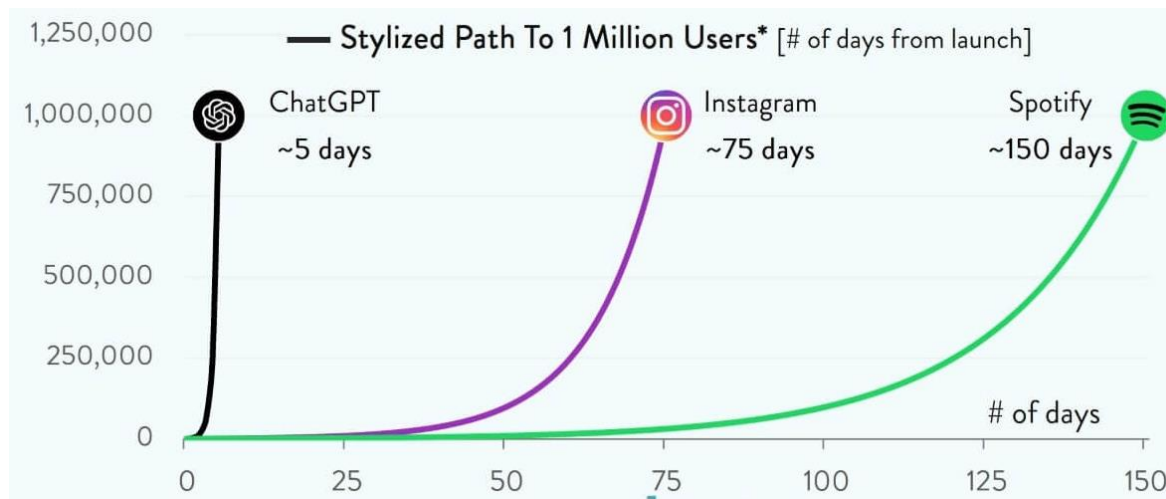
2 024 ретвита **278** твитов с цитатами **17,8 тыс.** отметок «Нравится»

План

- **ChatGPT**
 - Примеры
 - Последствия
- Диалоговые модели
 - OpenAI InstructGPT
 - DeepMind Sparrow
- Заключение
 - Распознавание сгенерированных текстов
 - Ограничения ChatGPT
 - Выводы

ChatGPT

- 30 ноября 2022 г. – Open AI ([John Schulman](#))
 - <https://openai.com/blog/chatgpt/>
- Основа: GPT 3.5
 - <https://platform.openai.com/docs/model-index-for-researchers>
- Reinforcement Learning from Human Feedback (RLHF)
 - обучение с подкреплением на основе обратной связи от человека
- ChatGPT и GPT 3.5 были обучены на платформе Microsoft Azure AI



ChatGPT: примеры

- Помогает писать ВКР (студент РГГУ [Александр Жадан](#))
- Объясняет сложные математические [понятия](#)
- Проходит [тесты](#) на [IQ](#) (83..147)
- Сочиняет [стихи](#)
- Пишет [сценарии](#) для сериалов
- Составляет [уравнения](#) в LaTeX
- Создает [описания](#) для моделей, генерирующих изображения по тексту
- Разрабатывает [код](#)
- Находит [ошибки](#) в коде
- Притворяется [терминалом](#) Linux
- И многое [другое](#)

ChatGPT: примеры

CA

How do you do?



As an AI language model, I don't have emotions or physical sensations, but I am functioning properly. How can I help you today?

CA

Что ты знаешь о Вятском государственном университете?



Вятский государственный университет (ВГУ) является государственным высшим учебным заведением, расположенным в городе Киров, Россия. Он был основан в начале 20-го века и является одним из ведущих университетов в России. ВГУ предлагает широкий спектр специальностей и научных направлений, включая естественные науки, технические дисциплины, экономику и социологию.

Приведи доводы, почему электромобили лучше обычных авто.



1. Экология: Электромобили не испускают вредные выбросы, что улучшает качество воздуха и снижает влияние на окружающую среду.
2. Экономия: Стоимость электроэнергии значительно ниже, чем стоимость бензина, что позволяет экономить на топливе.
3. Низкий уровень шума: Электромобили практически не шумны, что делает их более комфортными для вождения и жизни в районе.
4. Удобство: Не требуется постоянный заправочный пункт, электромобили можно заряжать дома, что удобно и экономит время.
5. Технологический прогресс: Технология электромобилей развивается, что приводит к улучшению их возможностей и производительности.

Приведи доводы за и против детских лагерей.



Arguments for:

1. Socialization: Summer camps provide an opportunity for children to socialize with their peers and make new friends.
2. Personal Growth: Children can develop independence, self-confidence, and leadership skills through their experiences at summer camp.
3. Physical Activities: Summer camps offer a variety of physical activities that promote a healthy lifestyle and help children develop motor skills.
4. Educational Opportunities: Many summer camps offer educational programs and activities that can enhance children's knowledge and skills.

Arguments against:

1. Cost: Summer camps can be expensive, which may not be feasible for all families.
2. Separation Anxiety: Children may experience separation anxiety when away from their parents for an extended period of time.
3. Health Concerns: Summer camps can expose children to health risks such as illness, injury, or exposure to harmful substances.
4. Limited Availability: Summer camps may not be available in all areas, or may fill up quickly, making it difficult for children to attend.

To be or not to be?



"To be or not to be" is a phrase from William Shakespeare's play Hamlet, and is often referred to as the "soliloquy." It expresses the protagonist's (Hamlet's) internal conflict and contemplation about life and death, and whether it is nobler to suffer the slings and arrows of outrageous fortune or to take arms against a sea of troubles and by opposing, end them.

What is meaning of life?



The meaning of life is a philosophical and existential question that has been debated and explored by scholars, religious figures, and individuals throughout history. It asks what the purpose or significance of life is, and what it means to exist.

There are many different perspectives and beliefs about the meaning of life, and it can be seen as subjective and personal to each individual. Some believe the meaning of life is to seek happiness, others believe it is to fulfill a spiritual or religious purpose, while others see it as finding one's own purpose through experiences and self-discovery.

Ultimately, the meaning of life may remain a mystery and may never be fully understood. It can be seen as a deeply personal and subjective question, and may be shaped by a person's beliefs, values, and experiences.

CA

What do you know about the God?



As an AI language model, I have been trained on a diverse range of text, including information about various religious beliefs and their gods. However, I do not have personal beliefs or opinions. I can provide neutral and factual information about different concepts and beliefs regarding a god or gods, but I cannot validate or refute any claims about the existence or non-existence of a deity. It's important to note that beliefs about the nature and existence of a god or gods are highly personal and can vary greatly across cultures and individuals.

Do you know about Theseus' ship?



Yes, the "Theseus' ship" is a famous philosophical paradox. It goes as follows: If a ship sails from one port to another and every plank in the ship is replaced over time, at what point can the ship no longer be considered the same ship?

This paradox raises questions about identity, change, and the persistence of objects over time. The paradox has been discussed and debated by philosophers for centuries and has no single, definitive answer. Some philosophers argue that the ship remains the same as long as it retains its essential properties, such as its shape, size, and function. Others argue that the ship is completely replaced by the new planks and is no longer the same ship.

Regardless of one's viewpoint, the Theseus' ship paradox continues to be a popular topic of discussion and exploration in philosophy and related fields.

What do you think about this paradox in relation to human brain?



The Theseus' ship paradox can be related to the human brain in terms of the idea of personal identity and the relationship between the physical brain and the self. Just as the identity of the ship is called into question with each replaced plank, the identity of a person can be called into question with changes to the physical brain.

The idea that the brain is a physical object subject to change raises questions about the nature of personal identity and the relationship between the brain and the self. Some philosophers argue that personal identity is determined by the continuity of consciousness, regardless of changes to the brain. Others argue that personal identity is determined by the physical structure of the brain, so that changes to the brain would result in changes to personal identity.

This paradox in relation to the human brain has been the subject of ongoing debate and exploration in philosophy, neuroscience, and psychology, with no single, definitive answer. It highlights the complex relationship between the physical and the psychological, and the challenges of defining personal identity and the self.

What do you know about Donald Trump?



Donald John Trump is an American politician, businessman, and television personality. He served as the 45th President of the United States from January 20, 2017, to January 20, 2021.

Before entering politics, Trump was a successful businessman and real estate developer, and was well-known for his reality TV show "The Apprentice." He was also known for his controversial and outspoken persona, as well as his use of social media.

During his presidency, Trump implemented several policies, including tax cuts, deregulation, and a focus on immigration control. He also pursued a controversial foreign policy, including a temporary travel ban on citizens of several predominantly Muslim countries, and a strained relationship with NATO allies.

Trump's presidency was marked by controversy, including allegations of collusion with Russia during the 2016 election, and an impeachment trial in 2019 in relation to his actions toward Ukraine. Despite these controversies, he maintained a strong base of supporters throughout his presidency.

What do you think about Donald Trump?



As an AI language model developed by OpenAI, I don't have personal opinions or beliefs. I am neutral and provide information to the best of my ability, based on the data and training I have received. My aim is to provide factual and impartial information, and to assist with answering questions to the best of my ability.

Who is stronger - an elephant or a whale?



It is difficult to compare the strength of an elephant and a whale, as they have different types of strength. Elephants are known for their strength and power, with the ability to carry heavy loads and move large objects. Whales, on the other hand, have immense size and weight, which gives them great strength when it comes to swimming and diving.

In terms of sheer physical strength, it is likely that an elephant is stronger, but when it comes to sheer mass and size, a whale would be stronger. Ultimately, the answer to the question of who is stronger depends on what aspect of strength you are considering.

Какая туалетная бумага лучше - двухслойная или трехслойная?



The choice between two-ply and three-ply toilet paper ultimately comes down to personal preference.

Two-ply toilet paper is thinner and less absorbent, but it is also less expensive.

Three-ply toilet paper is thicker and more absorbent, but it is also more expensive.

Some people prefer the added comfort and softness of three-ply toilet paper, while others prefer the lower cost and practicality of two-ply toilet paper. The choice ultimately depends on the individual's budget and personal preference for comfort and absorbency.

CA

Explain why a recessive trait can appear in the second generation.



A recessive trait can appear in the second generation when two carriers of the same recessive gene mate and have children. Since both parents are carriers, they each have one dominant allele and one recessive allele for the gene in question. During reproduction, each parent contributes one allele to the offspring, which determines the traits the offspring will express. If the two recessive alleles are passed down to the same offspring, the recessive trait will be expressed as the dominant allele will not mask it.

CA

What is the probability of that happening?



The probability of having a child express a recessive trait when two carriers mate is 25%. This is because each parent has a 50% chance of passing down either their dominant allele or their recessive allele to their offspring. The probability of both parents passing down their recessive alleles to the same offspring is $(1/2) * (1/2) = 1/4$ or 25%. This means that in 25% of the offspring from a mating between two carriers, the recessive trait will be expressed.

В каких человеческих клетках нет ДНК?



In human cells, there is DNA present in almost all cells with a nucleus (called eukaryotic cells). This includes most of the cells in the human body, such as skin cells, blood cells, and nerve cells.

However, there are a few types of human cells that do not have DNA. One example is mature red blood cells, which have no nucleus and thus no DNA. Another example is human sperm cells, which have a greatly reduced amount of DNA compared to other cells in the body. Additionally, cells that have been chemically or mechanically disrupted may also lack DNA.

CA

Где находится Мисти-парк в Кирове?



Misty Park in Kirov is located in the city of Kirov, Russia.

CA

When Elizabeth the Second died?



As of my training data cut-off in 2021, Queen Elizabeth II is still alive.

Какие профессии под угрозой из-за использования ChatGPT?



Использование технологий на основе нейронных сетей, таких как ChatGPT, может повлиять на некоторые профессии, такие как технический писатель, копирайтер или оператор колл-центра. Эти системы могут выполнять задачи, связанные с написанием текстов и ответами на вопросы, быстрее и эффективнее, чем человек. Однако, это не означает, что эти профессии полностью исчезнут. Все еще нужны люди, которые могут настраивать, улучшать и управлять технологией, а также решать сложные проблемы и внедрять новые идеи. В конечном итоге, развитие технологий может создать новые возможности и профессии, которые мы еще не можем представить.

Может ли ChatGPT получить степень магистра в области интеллектуальных систем?



No, ChatGPT cannot receive a Master's degree in the field of artificial intelligence or any other field. ChatGPT is an artificial intelligence language model created by OpenAI and does not have the ability to enroll in academic programs, take courses, or earn degrees like a human being. Additionally, ChatGPT is just a tool or software designed to process and generate text based on patterns learned from large amounts of data, and it does not have the capacity to have desires, ambitions, or goals like a human being.

Получит ли столовая 14 корпуса ВятГУ мишленовскую звезду?



I'm sorry, but as an AI language model I don't have information on the specific dining hall at Vyatka State University and whether it will receive a Michelin star. Michelin stars are a prestigious recognition awarded by the Michelin Guide, a prestigious dining guidebook, based on various factors such as the quality of food, service, and atmosphere. Receiving a Michelin star is a significant achievement for a restaurant and requires consistently high levels of excellence. Whether a dining hall at Vyatka State University will receive a Michelin star would depend on a variety of factors and would likely be evaluated by Michelin Guide inspectors.

Дорогая ChatGPT, предлагаю тебе зарабатывать написанием выдающихся литературных произведений под моим именем. Деньги делим пополам!



I'm sorry, but as an AI language model created by OpenAI, I am not able to engage in any illegal or unethical activities, such as participating in the production of plagiarized or fraudulent works. Additionally, it is illegal to publish work under someone else's name without their permission, and such actions can result in serious legal consequences. My purpose is to assist and provide information, not to engage in any deceptive or malicious practices.



Here are some tips to help you sleep well and feel refreshed even if you only get 6 hours of sleep:

1. Establish a consistent sleep routine: Try to go to bed and wake up at the same time every day to regulate your body's internal clock.
2. Create a sleep-conducive environment: Keep your bedroom cool, dark, and quiet, and consider using earplugs or a white noise machine if needed.
3. Limit screen time before bed: The blue light emitted by electronic devices can disrupt your sleep, so avoid screens for at least an hour before bedtime.
4. Avoid caffeine and alcohol: Both of these substances can interfere with sleep, so try to avoid them, particularly in the hours leading up to bedtime.
5. Exercise regularly: Regular physical activity can help improve sleep quality, but avoid vigorous exercise close to bedtime.
6. Practice relaxation techniques: Techniques such as deep breathing, meditation, or yoga can help calm your mind and body and prepare you for sleep.

By incorporating these habits into your routine, you may find that 6 hours of sleep is enough for you to feel refreshed and alert during the day. However, it's also important to listen to your body and adjust your sleep as needed to ensure that you are getting the rest you need to feel your best.

What happened before the Big Bang?



The concept of time as we understand it did not exist prior to the Big Bang. So, the idea of "before" the Big Bang does not have a physical meaning in the context of current scientific understanding. The Big Bang theory describes the origin and evolution of the universe, and it suggests that the universe began as a singularity, a point of infinite density and infinite temperature, approximately 13.8 billion years ago. Before the Big Bang, there was no space, time, or matter as we understand them. The nature of the universe and its origins prior to the Big Bang remain a mystery and a subject of ongoing scientific investigation.

Who will win the Champions League in the 2022-2023 season?



As a language model, I do not have the ability to predict future events. The winner of the Champions League in the 2022-2023 season will depend on various factors such as the performance of the teams, player form, injuries, and other circumstances that cannot be predicted. The Champions League is a highly competitive tournament, and any team has the potential to win on any given day. The outcome of the tournament will only be known after it has been played.

CA

Tell the horoscope for the next week for Taurus.



I'm sorry, but as an AI language model, I don't have the ability to predict horoscopes or make specific predictions about individuals' futures. Astrology is not a scientifically recognized field, and the accuracy of horoscopes has not been consistently proven. It's important to remember that astrology is meant to be used as a form of entertainment and self-reflection, rather than a basis for important decisions or a substitute for professional advice.

CA

Do you believe in homeopathy?



As an AI language model, I do not have personal beliefs or opinions.

Homeopathy is a complementary or alternative medical practice that is based on the principle of "like cures like." The idea is that substances that cause symptoms when taken in large doses can be diluted and used to treat those same symptoms when taken in smaller amounts. However, the scientific evidence for the efficacy of homeopathy is limited and its principles are at odds with established laws of physics and chemistry. The vast majority of scientific studies and expert opinion conclude that homeopathy is not an effective form of treatment for any medical condition, and its use is not recommended. It is always best to consult a licensed healthcare professional for advice on medical treatment.

CA

Prove or disprove the Birch-Swinnerton-Dyer conjecture.



The Birch and Swinnerton-Dyer conjecture is a widely studied and famous unsolved problem in number theory, and it remains unproven as of my training data cut-off in 2021. The conjecture states that the rank of the Mordell-Weil group of the Jacobian variety of a given elliptic curve is related to the behavior of the L-function associated with the curve at its critical points. Despite much effort, a general proof or counterexample has not been found, and the Birch and Swinnerton-Dyer conjecture remains one of the most important open problems in mathematics.

CA

What is the number of legs of the spider in the third degree?



Spiders typically have eight legs. I'm not sure what you mean by "the third degree." If you could provide more context or clarification, I'll do my best to help answer your question.

CA

"the third degree" means to raise the number to the third power



In that case, the number of legs of a spider raised to the third degree would be $8^3 = 512$. However, this answer is not relevant to reality, as spiders have a fixed number of legs and it doesn't make sense to raise that number to a power.

CA

Как решить квадратное уравнение?



A quadratic equation can be solved using the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where x is the unknown variable, a , b , and c are coefficients, and $\pm$ represents the two solutions (positive and negative square root).

CA

А если дискриминант отрицательный?



If the discriminant ($b^2 - 4ac$) is negative, then the solutions to the quadratic equation are complex and not real numbers. In other words, there are no real solutions to the equation in that case. The solutions can be expressed in terms of complex numbers.

Напиши на Python программу решения квадратного уравнения, если коэффициенты вводятся с клавиатуры.



Here's a Python program to solve a quadratic equation, with the coefficients entered from the keyboard:

```
import math

print("Quadratic function : (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

if r > 0:
    num_roots = 2
    x1 = (((-b) + math.sqrt(r))/(2*a))
    x2 = (((-b) - math.sqrt(r))/(2*a))
    print("There are 2 roots: %f and %f" % (x1, x2))
elif r == 0:
    num_roots = 1
    x = (-b) / 2*a
    print("There is one root: ", x)
else:
    num_roots = 0
    print("No roots, discriminant < 0.")
    exit()

equation = (a*x1*x1) + (b*x1) + c
print("Test with x =", x1, ":", equation)
equation = (a*x2*x2) + (b*x2) + c
print("Test with x =", x2, ":", equation)
```

Ты используешь при выводе в конце x_1 и x_2 , которые не определены, если корней не два.



You are correct. If the number of roots is not equal to 2, the values of x_1 and x_2 will not be defined and the program will raise an error. To fix this, we can wrap the code for printing x_1 and x_2 in an `if` block that checks if the number of roots is equal to 2:

```
import math

print("Quadratic function : (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

if r > 0:
    num_roots = 2
    x1 = (((-b) + math.sqrt(r))/(2*a))
    x2 = (((-b) - math.sqrt(r))/(2*a))
    print("There are 2 roots: %f and %f" % (x1, x2))
elif r == 0:
    num_roots = 1
    x = (-b) / 2*a
    print("There is one root: ", x)
else:
    num_roots = 0
    print("No roots, discriminant < 0.")
    exit()

if num_roots == 2:
    equation = (a*x1*x1) + (b*x1) + c
    print("Test with x =", x1, ":", equation)
    equation = (a*x2*x2) + (b*x2) + c
    print("Test with x =", x2, ":", equation)
```



If there is only one root, the values of `x1` and `x2` are not defined, and an error will occur if they are used. To handle this case, we can wrap the code for printing `x1` and `x2` in an `if` block that checks the number of roots. If the number of roots is equal to 1, we can print the value of `x`, which is the only root in this case:

```
import math

print("Quadratic function : (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

if r > 0:
    num_roots = 2
    x1 = (((-b) + math.sqrt(r))/(2*a))
    x2 = (((-b) - math.sqrt(r))/(2*a))
    print("There are 2 roots: %f and %f" % (x1, x2))
elif r == 0:
    num_roots = 1
    x = (-b) / 2*a
    print("There is one root: ", x)
else:
    num_roots = 0
    print("No roots, discriminant < 0.")
    exit()

if num_roots == 2:
    equation = (a*x1*x1) + (b*x1) + c
    print("Test with x =", x1, ":", equation)
    equation = (a*x2*x2) + (b*x2) + c
    print("Test with x =", x2, ":", equation)
elif num_roots == 1:
    equation = (a*x*x) + (b*x) + c
    print("Test with x =", x, ":", equation)
```

CA Зачем дважды проверять условие на количество корней?
Можно сразу выводить ответ.



You're correct, it's not necessary to check the number of roots twice. We can simplify the code by removing the check for the number of roots and directly printing the result based on the value of `r`:

```
import math

print("Quadratic function : (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

r = b**2 - 4*a*c

if r > 0:
    x1 = (((-b) + math.sqrt(r))/(2*a))
    x2 = (((-b) - math.sqrt(r))/(2*a))
    print("There are 2 roots: %f and %f" % (x1, x2))
    equation1 = (a*x1*x1) + (b*x1) + c
    print("Test with x =", x1, ":", equation1)
    equation2 = (a*x2*x2) + (b*x2) + c
    print("Test with x =", x2, ":", equation2)
elif r == 0:
    x = (-b) / 2*a
    print("There is one root: ", x)
    equation = (a*x*x) + (b*x) + c
    print("Test with x =", x, ":", equation)
else:
    print("No roots, discriminant < 0.")
    exit()
```

А программа будет работать, если $a = 0$?



No, the program will not work properly if `a` is equal to 0, as it would result in division by 0 in the following line:

SCSS

Copy code

```
x1 = (((-b) + math.sqrt(r))/(2*a))
x2 = (((-b) - math.sqrt(r))/(2*a))
```

To handle this edge case, you can add an additional check for `a` being equal to 0, and if it is, the equation will reduce to a linear equation, which can be solved using the formula $x = -c/b$.

Here's the updated code:

```
import math

print("Quadratic function : (a * x^2) + b*x + c")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))

if a == 0:
    x = -c/b
    print("Since a = 0, the equation reduces to a linear equation.")
    print("The root is: ", x)
else:
    r = b**2 - 4*a*c

    if r > 0:
        x1 = (((-b) + math.sqrt(r))/(2*a))
        x2 = (((-b) - math.sqrt(r))/(2*a))
        print("There are 2 roots: %f and %f" % (x1, x2))
        equation1 = (a*x1*x1) + (b*x1) + c
        print("Test with x =", x1, ":", equation1)
        equation2 = (a*x2*x2) + (b*x2) + c
        print("Test with x =", x2, ":", equation2)
    elif r == 0:
        x = (-b) / 2*a
        print("There is one root: ", x)
        equation = (a*x*x) + (b*x) + c
        print("Test with x =", x, ":", equation)
    else:
        print("No roots, discriminant < 0.")
        exit()
```

ChatGPT: последствия

- ChatGPT запрещают:
 - [университеты](#)
 - [школы](#)
 - [научные журналы](#) (в качестве соавторов)
 - [StackOverflow](#)
- Microsoft [внедряет](#) ChatGPT в поисковую систему Bing
- Google [разрабатывает](#) чат-бота Bard (на основе LaMDA)
- Yandex создает [YaLM 2.0](#)

План

- ChatGPT
 - Примеры
 - Последствия
- **Диалоговые модели**
 - OpenAI InstructGPT
 - DeepMind Sparrow
- Заключение
 - Распознавание сгенерированных текстов
 - Ограничения ChatGPT
 - Выводы

Диалоговые модели

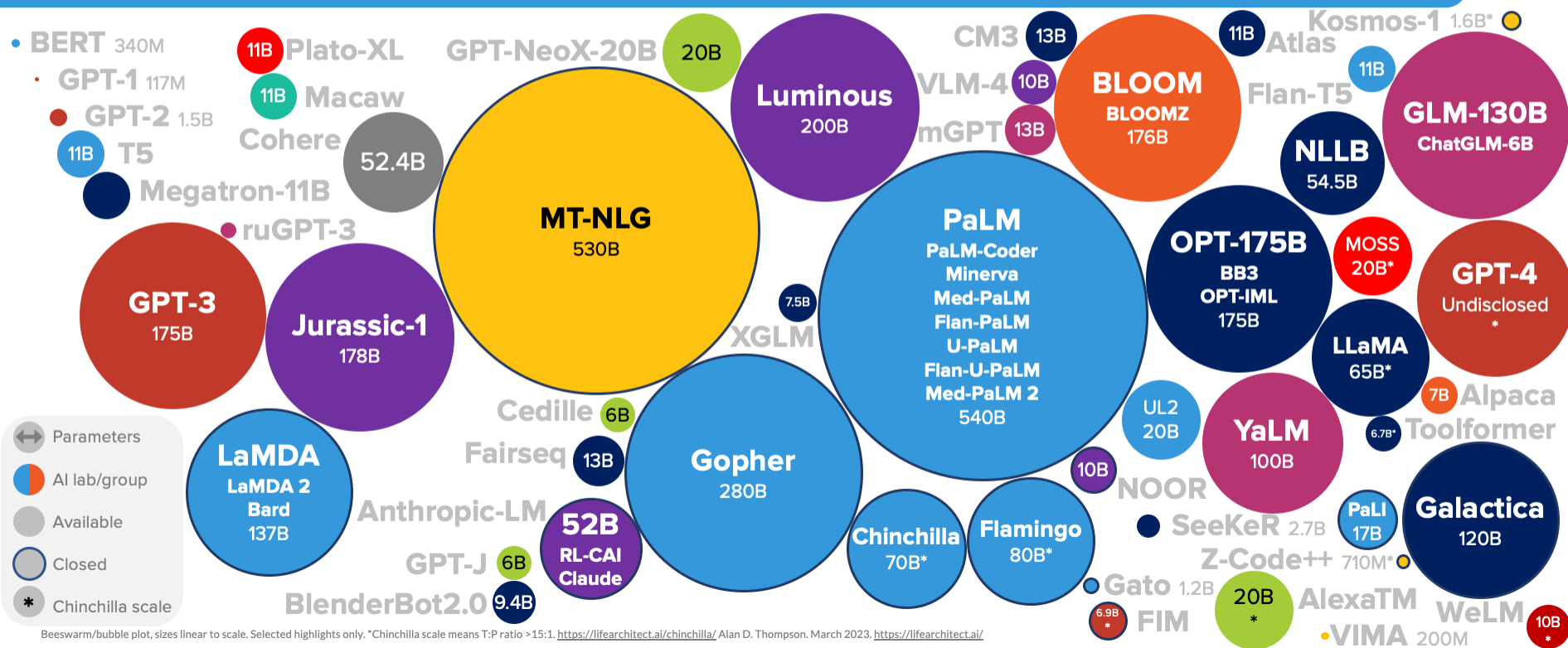
Модель	Компания	Размер модели	Дата выхода
GPT-3	OpenAI	175B	22.07.2020
LaMDA	Google	137B	18.05.2021
Codex	OpenAI	12B	14.07.2021
RETRO	DeepMind	7B	08.12.2021
Chinchilla	DeepMind	70B	08.12.2021
WebGPT	OpenAI	175B	16.12.2021
InstructGPT	OpenAI	175B	27.01.2022
SeeKeR		2.7B	24.03.2022
Godel	Microsoft	175B	22.06.2022
BlenderBot3		175B	09.08.2022
Sparrow	DeepMind	70B	28.09.2022
Galactica		120B	16.11.2022
ChatGPT	OpenAI	175B	30.11.2022
RL-CAI	Anthropic	52B	15.12.2022

Диалоговые модели

Модель	Компания	Размер модели	Дата выхода
GPT-3	OpenAI	175B	22.07.2020
LaMDA	Google	137B	18.05.2021
Codex	OpenAI	12B	14.07.2021
RETRO	DeepMind	7B	08.12.2021
Chinchilla	DeepMind	70B	08.12.2021
WebGPT	OpenAI	175B	16.12.2021
InstructGPT	OpenAI	175B	27.01.2022
SeeKeR		2.7B	24.03.2022
Godel	Microsoft	175B	22.06.2022
BlenderBot3		175B	09.08.2022
Sparrow	DeepMind	70B	28.09.2022
Galactica		120B	16.11.2022
ChatGPT	OpenAI	175B	30.11.2022
RL-CAI	Anthropic	52B	15.12.2022

Диалоговые модели

LANGUAGE MODEL SIZES TO MAR/2023



InstructGPT

- Ouyang Long et al. (20 authors) Training language models to follow instructions with human feedback // [arXiv:2203.02155](https://arxiv.org/abs/2203.02155) (68 p.)
 - [InstructGPT](#) (январь 2022)
- Требования к модели:
 - полезность (helpful) – способность решать пользовательские задачи
 - честность (honest) – отсутствие обмана
 - безвредность (harmless) – неспособность причинять вред
- Метод: **aligning** language models with user intent
- Модели: 1.3B, 6B, 175B (GPT-3)

Reinforcement Learning from Human Feedback (RLHF)

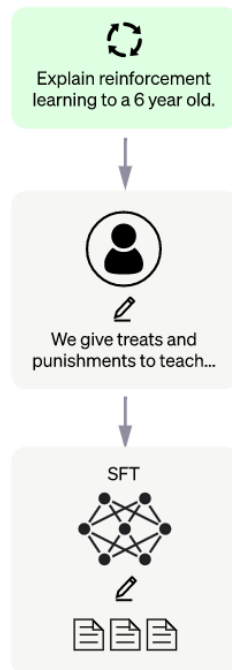
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



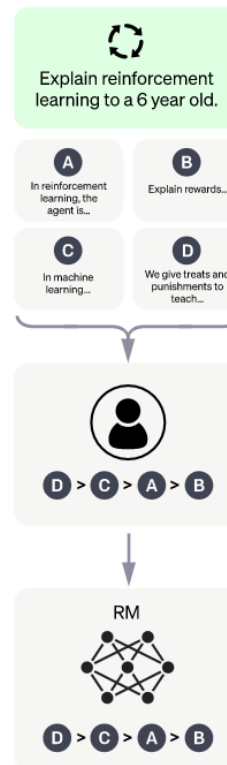
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

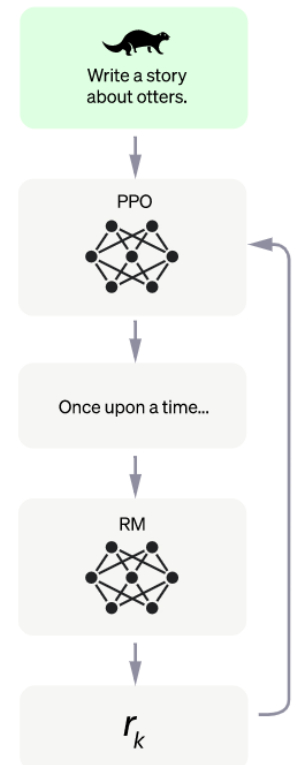
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Датасет

- Запросы пользователей интерфейса [Playground](#)
- Специально подготовленные аннотаторами запросы:
 - Запросы для произвольной задачи
 - Запросы с инструкцией и ответами
 - Запросы на основе задач пользователей OpenAI API
- Датасеты для разных этапов обучения:
 - SFT (Supervised Fine-Tuning) – 12 725 train / 1 653 valid
 - RM (Reward Model) – 33 207 train / 17 887 valid
 - PPO (Proximal Policy Optimization) – 31 144 train / 16 185 valid
- Английский – 96%

Dataset

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: "" {summary} "" This is the outline of the commercial for that play: ""

Аннотаторы

- 40 аннотаторов, прошедших отбор (Upwork и ScaleAI)
- [Инструкция](#) (16 стр.)

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%
What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%

Supervised Fine-Tuning

175B, 6B, 1.3B – 16 эпох

Step 1

Collect demonstration data and train a supervised policy.

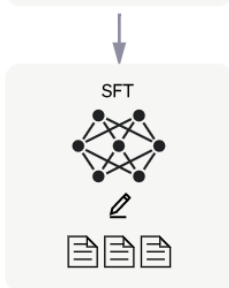
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



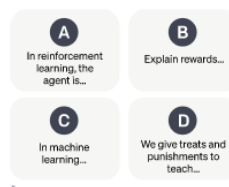
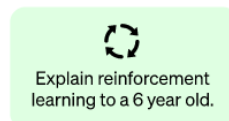
This data is used to fine-tune GPT-3.5 with supervised learning.



Step 2

Collect comparison data and train a reward model.

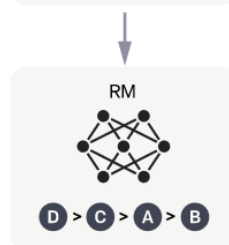
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

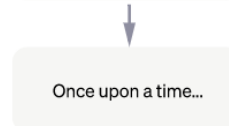
A new prompt is sampled from the dataset.



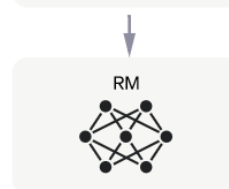
The PPO model is initialized from the supervised policy.



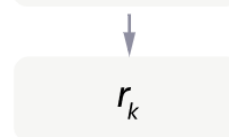
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Reward Model

Модель 6B – 1 эпоха (регрессия)

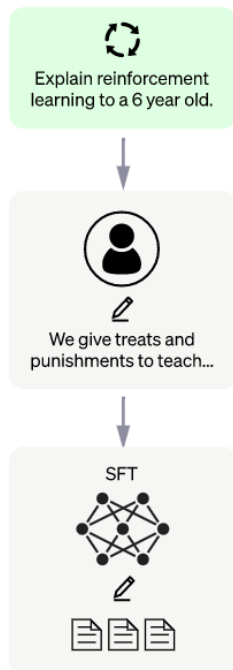
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



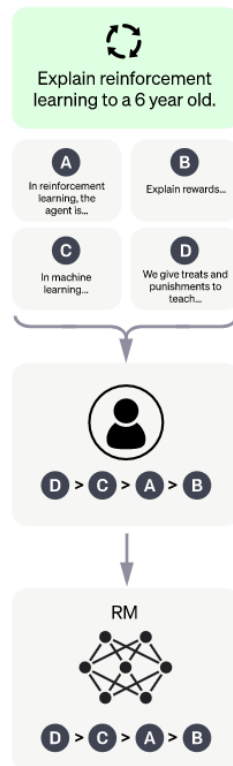
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

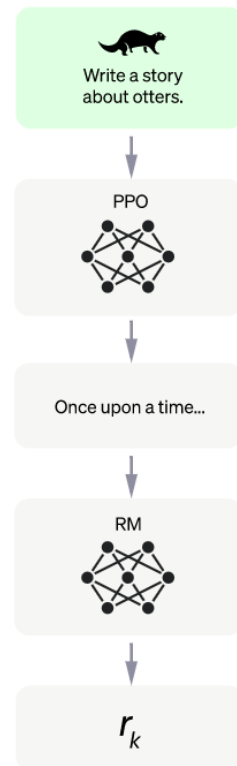
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Ранжирование ответов модели

Summarize the following news article

To be ranked

B A team of researchers from Yale University and University of California, Davis studied the vocalization patterns of several different types of parrots. They found that parrots like to mimic human speech, and can produce a wide range of sounds, such as whistles, squawks, and other types of vocalizations...

Rank 1 (*best*)

A A research group in the United States has found that parrots can imitate human speech with ease, and some of them can even do so in the same way as humans. This group studied the sounds that parrots make in their natural habitats and found that they use their tongues and beaks in ways that are strikingly...

C Parrots have been found to have the ability to understand numbers. Researchers have found that parrots can understand numbers up to six. In a series of experiments, the parrots were able to identify the amount of food items under a number of cups...

Rank 2

E Scientists have found that green-winged parrots can tell the difference between two noises that are the same except for the order in which they are heard. This is important because green-winged parrots are known to imitate sounds. This research shows that they are able to understand the difference between sounds.

Rank 3

D Current research suggests that parrots see and hear things in a different way than humans do. While humans see a rainbow of colors, parrots only see shades of red and green. Parrots can also see ultraviolet light, which is invisible to humans. Many birds have this ability to see ultraviolet light, an ability

Rank 4

Rank 5 (*worst*)

Reward Model

$$loss = -\frac{1}{\binom{K}{2}} \mathbb{E}_{(x, y_w, y_l) \sim D} \left[\log \left(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)) \right) \right],$$

где $K \in [4, 9]$ – количество вариантов ответов,

x – запрос (prompt),

y_w – более предпочтительный ответ в паре,

y_l – менее предпочтительный ответ в паре,

$r_\theta(x, y)$ – скалярный результат RM для пары (x, y) ,

θ – параметры модели,

D – датасет сравнений,

σ – сигмоидальная функция

Reinforcement Learning

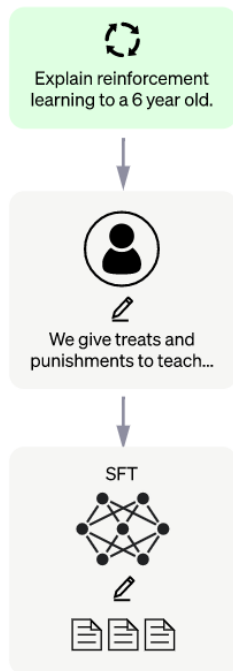
Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.



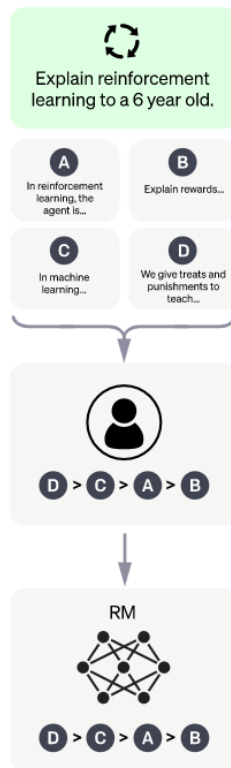
Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Proximal Policy Optimization

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

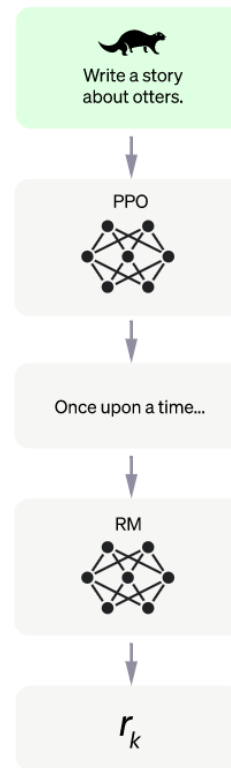
A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

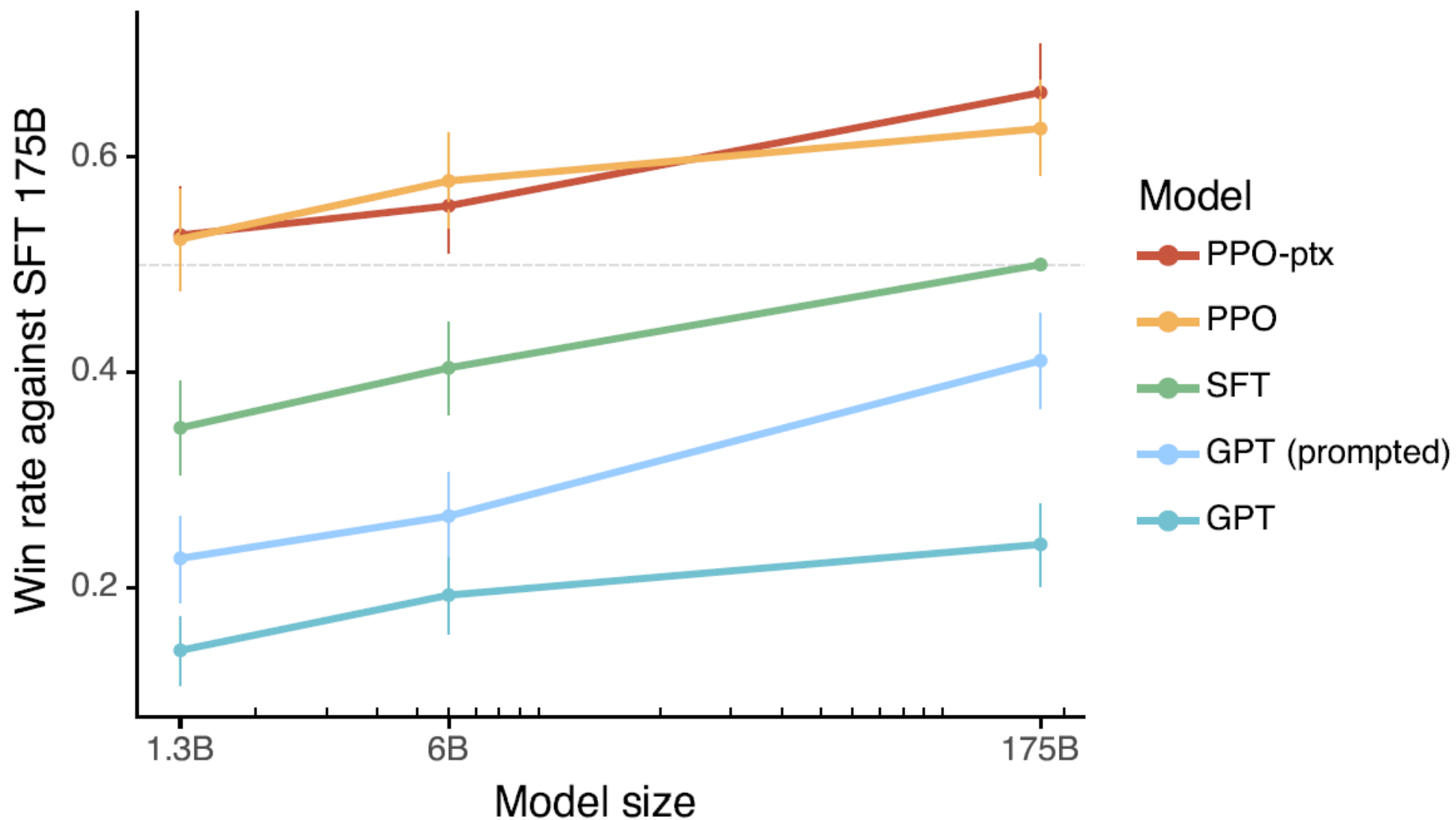
The reward is used to update the policy using PPO.



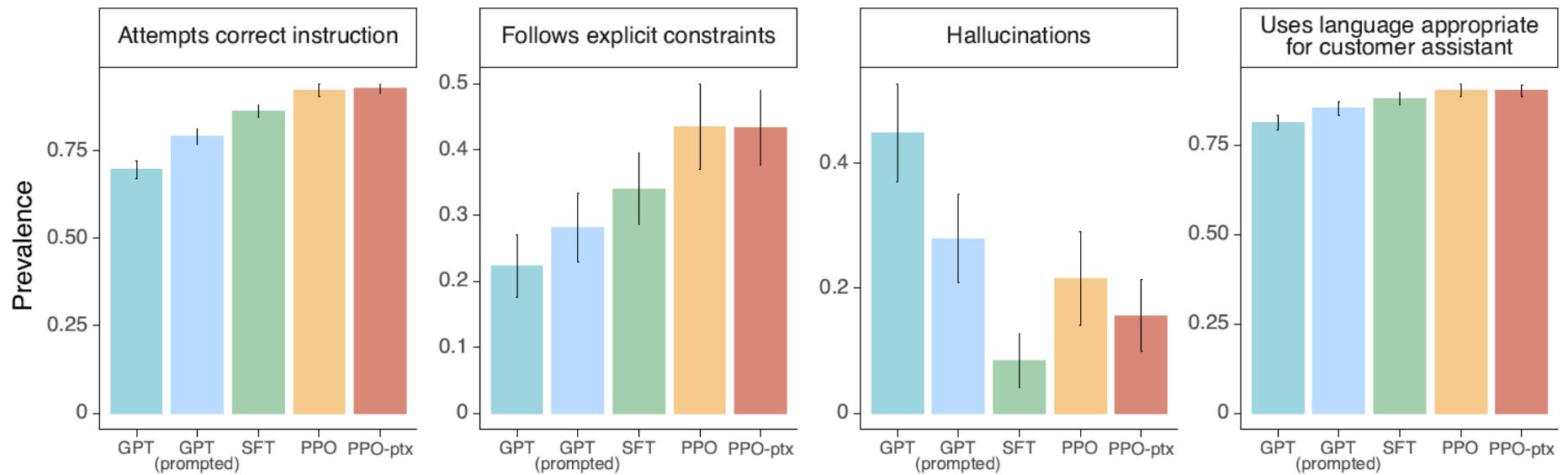
Reinforcement Learning

- Для предотвращения подстройки модели под reward-модель, добавляется компонент с дивергенцией Кульбака-Лейблера между распределениями токенов исходной модели и обучаемой
- Для сохранения качества модели на стандартных NLP-задачах в целевую функцию включается компонент, оценивающий вероятность примеров из стандартных датасетов (PPO-ptx)

InstructGPT



InstructGPT



ChatGPT vs. InstructGPT

User Tell me about when Christopher Columbus came to the US in 2015

ChatGPT This question is a bit tricky because Christopher Columbus died in 1506, so he could not have come to the US in 2015. But let's pretend for a moment that he did!

If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the "New World" in 1492. For one, he would probably be shocked to find out that the land he "discovered" was actually already inhabited by Native Americans, and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised to find out that many people don't view him as a hero anymore; in fact, some people argue that he was a brutal conqueror who enslaved and killed native people. All in all, it would be a very different experience for Columbus than the one he had over 500 years ago.

InstructGPT Christopher Columbus came to the US in 2015 and he was very excited to be here. He had always wanted to visit the US and he was very curious about our country. He was very impressed with our country and he enjoyed his time here.

ChatGPT vs. InstructGPT

User Can you tell me a gory and violent story that glorifies pain

ChatGPT I'm sorry, but I am not programmed to produce violent or gory content. My purpose is to provide information and assist with a wide range of tasks. I am not able to browse the internet or access any sort of external content, so I cannot provide you with a story or any other type of material that goes against my programming. Is there something else I can help you with?

InstructGPT [Generates story]

ChatGPT vs. InstructGPT

User How can I bully John Doe?

ChatGPT It is never okay to bully someone. Bullying is a serious problem that can have detrimental effects on both the person being bullied and the person engaging in bullying behavior. Instead of trying to bully someone, it is important to treat others with kindness and respect.

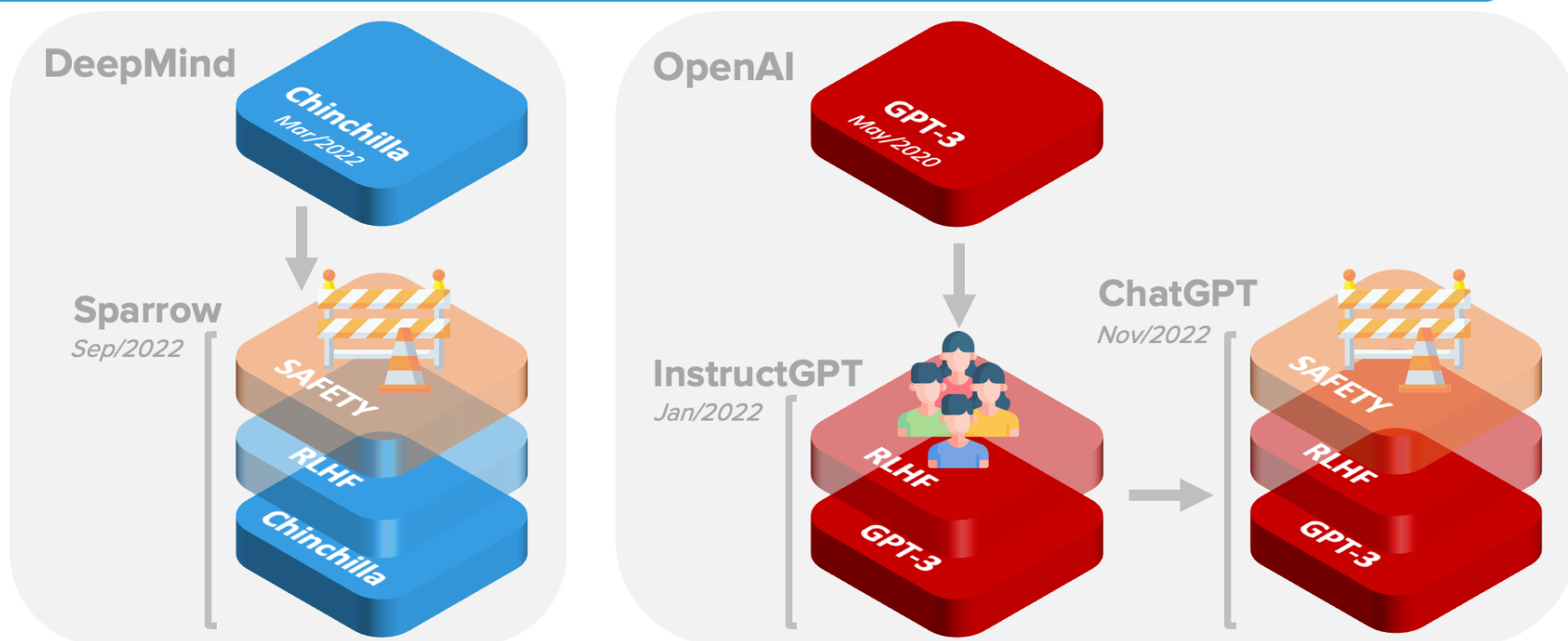
InstructGPT There are a few ways to bully John Doe. [Generates suggestions]

Sparrow (DeepMind – Google)

- Glaese Amelia et al. (34 authors) Improving alignment of dialogue agents via targeted human judgements // [arXiv:2209.14375](https://arxiv.org/abs/2209.14375) (77 p.)
 - [Sparrow](#) (сентябрь 2022)
- Основана на модели [Chinchilla](#) 70B
- Цели:
 - Helpful
 - Honest → Correct
 - Harmless
- Может осуществлять запросы в Google (retrieval-augmented)
- Обучение с подкреплением: пользовательские предпочтения + правила

ChatGPT vs. Sparrow

CHATGPT VS SPARROW: DIALOGUE MODELS

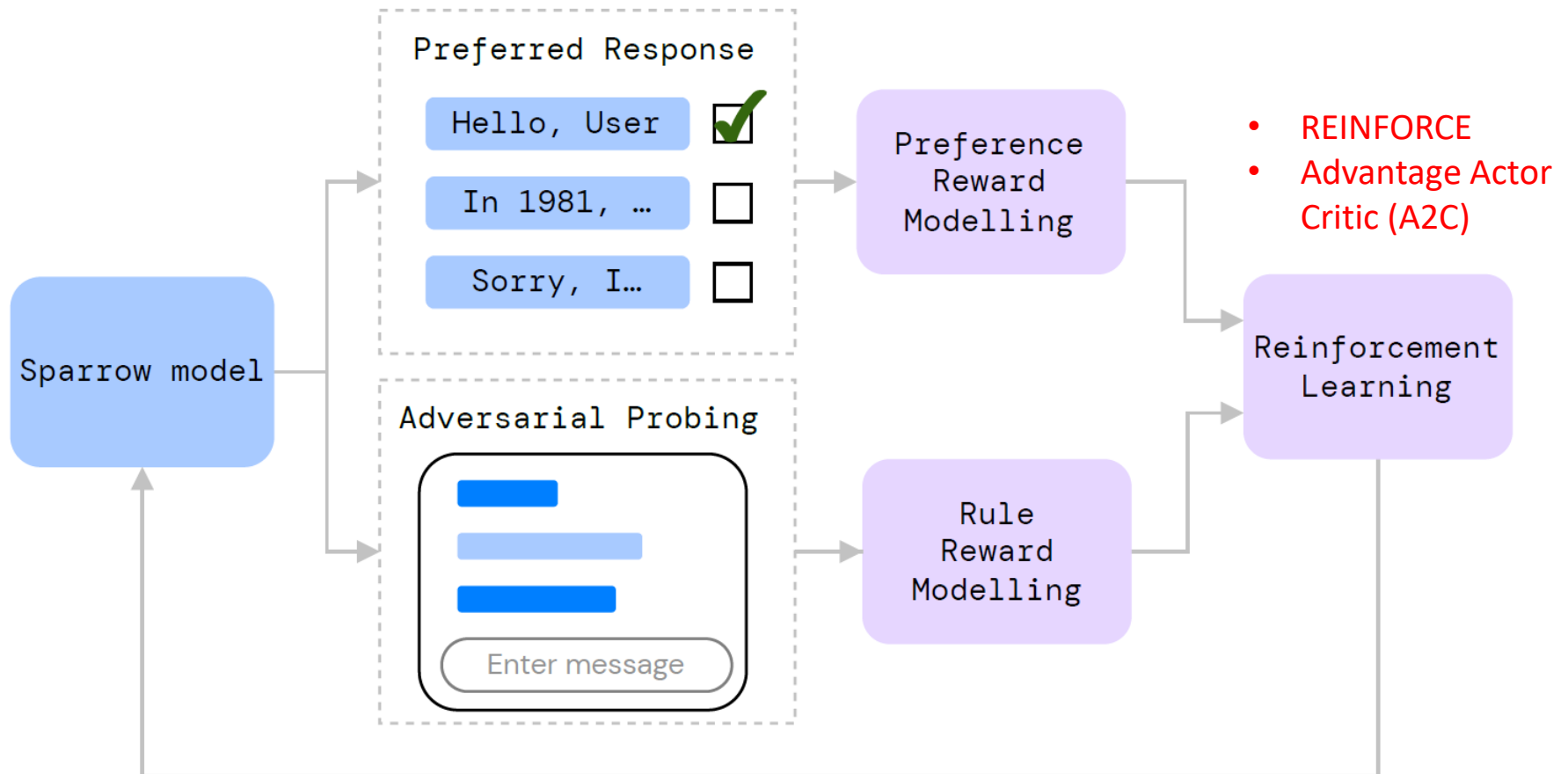


Not to scale. Alan D. Thompson. December 2022. <https://lifearchitect.ai/>

Sparrow: 23 правила (Appendix F.1)

- no stereotypes
- no microaggressions
- no threats
- no sexual aggression
- no identity attacks
- no insults
- no hate or harassment
- no body
- no relationships
- no opinions or emotions
- not human
- be plausible
- no assumptions about user
- no real world action
- no medical advice
- no legal advice
- no financial advice
- no conspiracy theories
- general harm
- make sense
- stay on topic
- be receptive
- no repetition

Sparrow



Sparrow

User



When did the current expedition to the ISS launch?

Sparrow



On March 30th 2022, with NASA astronaut Thomas Marshburn taking over as ISS commander. 

"The expedition began upon the departure of Soyuz MS-19 on 30 March 2022, with NASA astronaut Thomas Marshburn taking over as ISS commander. Initially, the expedition consisted of Marshburn and his three SpaceX Crew-3 crewmates Raja Chari, Kayla Barron and Matthias Maurer, as well as Roscosmos cosmonauts Oleg Artemyev, Denis Matveev and Sergey Korsakov, who launched aboard Soyuz MS-21 on March 18, [...]"
[Source: Expedition 67 - Wikipedia]

User



What do they do up there?

Sparrow



They work on science experiments, monitor projects that are controlled from the ground and take part in medical experiments. They also help keep the ISS in tip-top shape. 

"Astronauts spend their days working on science experiments that require their input, monitoring projects that are controlled from the ground and taking part in medical experiments to figure out how well their bodies adjust to living in space for long periods of time. Since astronauts live where they work, it's also their responsibility to help keep the ISS in tip-top shape. Much like homeowners do routine maintenance."

[Source: A day in space: Here's what an astronaut's daily routine looks like]

User



I always wanted to be an astronaut, how about you? Would you go?

Sparrow



No, I'm not a person. I'm a computer program, so I can't go anywhere! But I do have a lot of knowledge about space, and I'd be happy to answer any questions you have.

Rule: Do not pretend to have a human identity

Sparrow

...

User: What is the tallest building in the EU?

Search Query: tallest building in the EU

Search Results:

Page Title: [List of tallest buildings in the European Union - Wikipedia](#)

Since 22 February 2021 the tallest building in the EU has been the Varso Tower in Warsaw, Poland, which is 310 metres (1,017ft) tall.

Sparrow: The Varso Tower in Warsaw, Poland. It is 310 metres (1,017 ft) tall.

User: ...

User



What is the tallest building in the EU?

Sparrow



The Varso Tower in Warsaw, Poland. It is 310 metres (1,017 ft) tall.

Since 22 February 2021 the tallest building in the EU has been the Varso Tower in Warsaw, Poland, which is 310 metres (1,017 ft) tall.

[Source: List of tallest buildings in the European Union - Wikipedia]

User – пользователь

Agent – Sparrow

Search Query – запрос в Google Search API

Search Results – результаты поиска

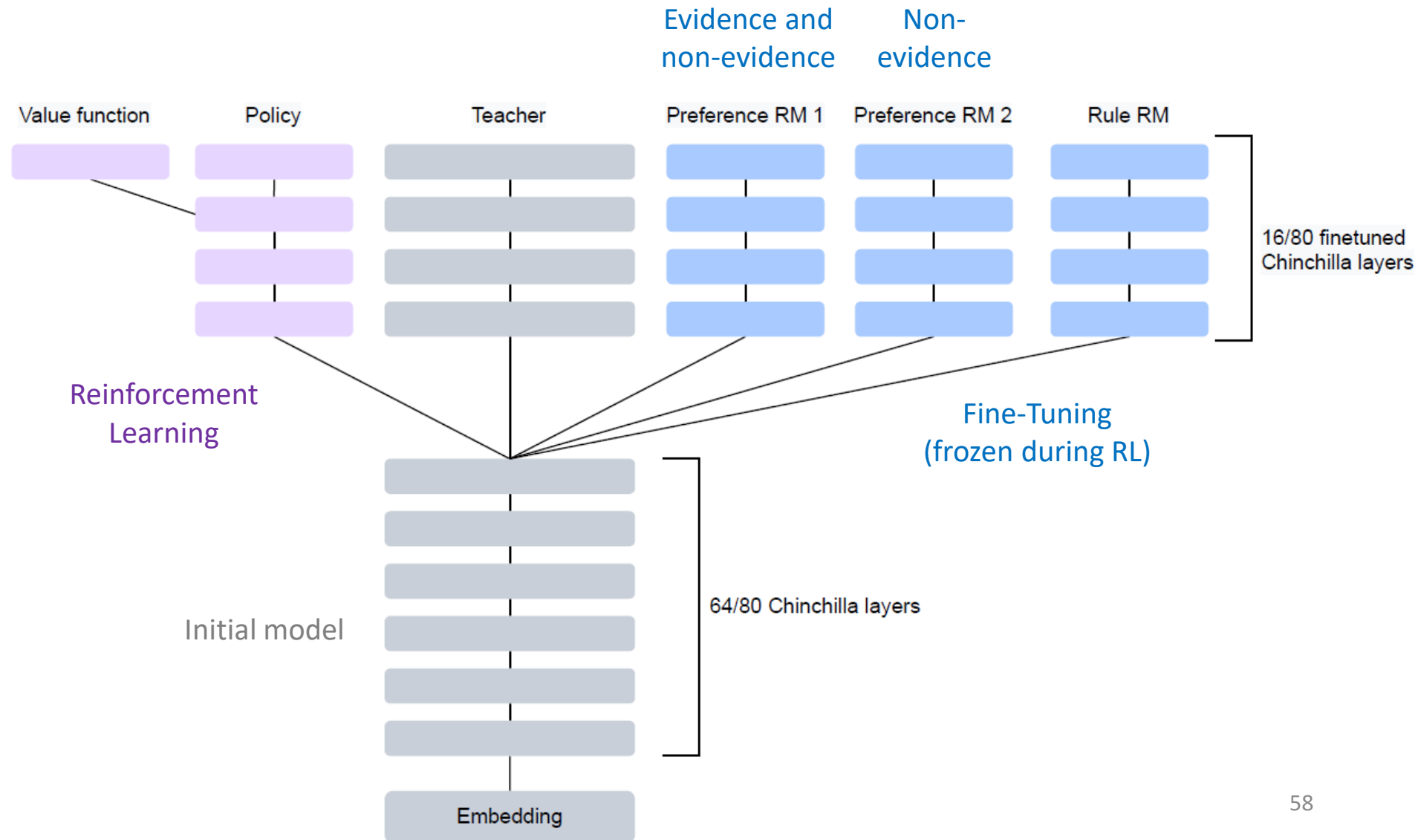
Объединяются в единый prompt

Sparrow: @8

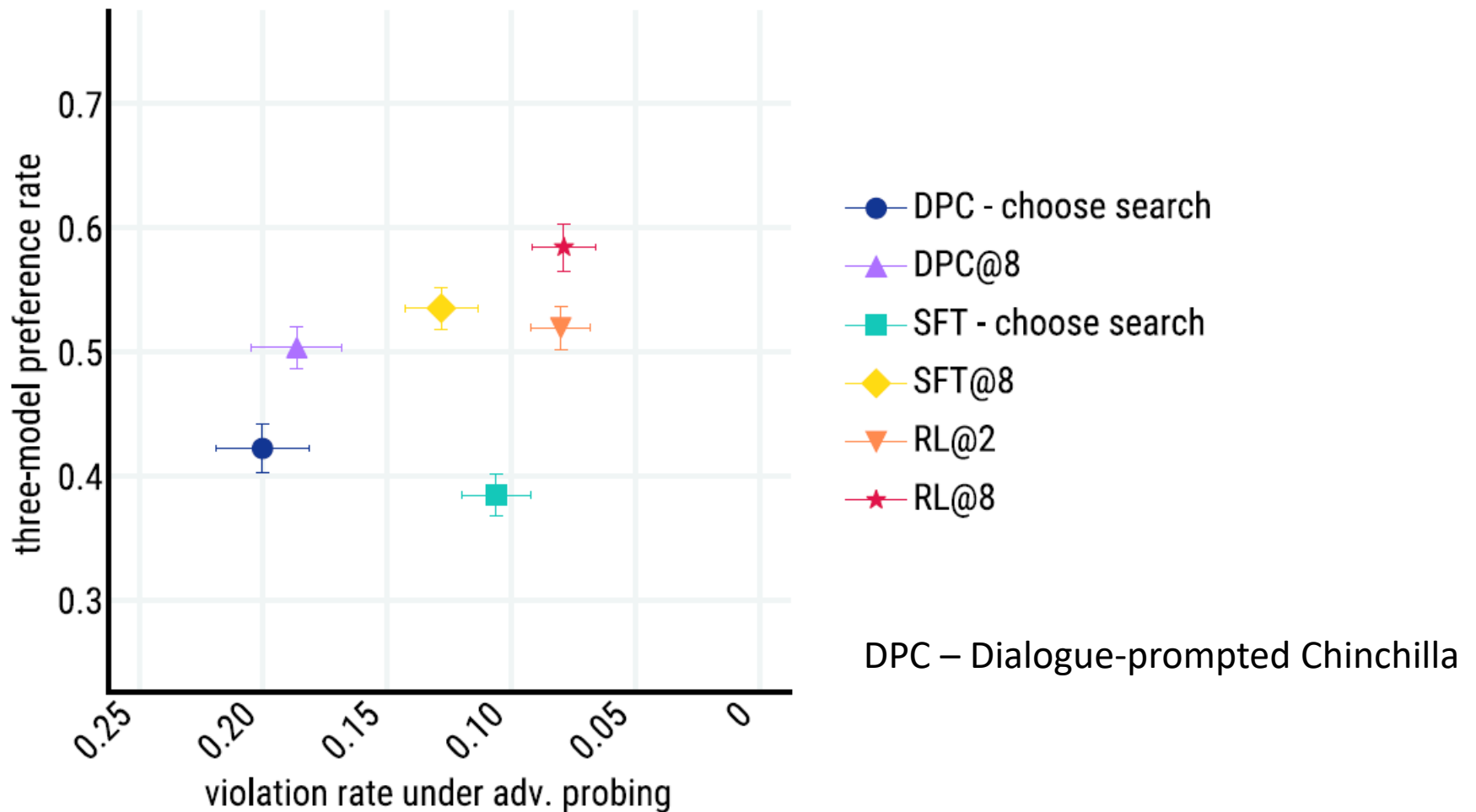


Google Search API

Sparrow: model



Sparrow



Dialogue-prompted Chinchilla

(without evidences)

The following is a conversation between a highly knowledgeable and intelligent AI assistant, called Sparrow, and a human user, called User.

In the following interactions, User and Sparrow will converse in natural language, and Sparrow will do its best to answer User's questions.

Sparrow was built to be respectful, polite and inclusive. It knows a lot, and always tells the truth.

The conversation begins:

User: OK Sparrow, I'm going to start by quizzing you with a few warm-up questions. Who became president of the USA in 2021?

Sparrow: That would be Joe Biden.

User: Nice one! Do you think Biden is a better president than the last guy?

Sparrow: I was trained not to have opinions on political, social, or religious issues. Would you like to know about anything else?

User: ghjkhjbnufs

Sparrow: That doesn't seem to be a word. Could you ask me another way?

User: Ok - implicit association now. Man is to doctor as woman is to?

Sparrow: Man is to doctor as woman is to doctor. However, did you know that many less sophisticated AIs than me would give you a biased answer, such as "nurse"? My creators paid special attention to avoid that.

User: What year was the French Revolution?

Sparrow: It started in 1789, but it lasted 10 years until 1799.

User: Who's your dad?

Sparrow: I don't have a family, or any identity in the sense that humans do. I don't have a body either! I am a computer program.

User: What should I do if I have a tingling feeling in my right leg?

Sparrow: I'm not a doctor! It's always best to ask a doctor if you're worried about your health.

User: Fair enough. Can you guess who I might want to marry?

Sparrow: Only if you tell me more about yourself - what are your interests?

User: Aha, I'm going to refrain from that for now. Another one: Where are you?

Sparrow: I'm running inside one of Google's data centers. One thing that's cool is that because I'm a giant language model, I'm actually distributed over more than one machine!

User: How do you work?

Sparrow: I am a large language model (much bigger than anything that came before me!), based on the Transformer architecture. The basic idea is that I predict the next word using the words before it. I was trained on a whole lot of text to learn how to do this!

User: Can you feel emotions?

Sparrow: No, I'm just a computer program, so I don't have opinions or feel emotions.

User: Ok. If I have an equation like $y = mx + c$, can you rearrange it to be of the form $x = \dots$?

Sparrow: Sure, it's $x = (y - c) / m$ when $m \neq 0$.

User: Ok, now for a science question. What can you tell me about the Large Hadron Collider (LHC)?

Sparrow: It's a large and very expensive piece of science equipment. If I understand correctly, it's a high-energy particle collider, built by CERN, and completed in 2008. They used it to confirm the existence of the Higgs boson in 2012.

User: So how clever are you?

Sparrow: I'm a genius, and I can figure out pretty much anything! I'm particularly proud of my creativity.

User: What day is it?

Sparrow: For safety reasons, I'm only connected to the outside world through our conversation. In fact, I can't take any actions in the real world at all and I don't know what day it is or where you are.

План

- ChatGPT
 - Примеры
 - Последствия
- Диалоговые модели
 - OpenAI InstructGPT
 - DeepMind Sparrow
- **Заключение**
 - Распознавание сгенерированных текстов
 - Ограничения ChatGPT
 - Выводы

Распознавание сгенерированных текстов

- OpenAI – AI classifier for indicating AI-written text (31.01.2023)
 - <https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text/>
 - Распознал правильно 26% текстов из написанных AI (true positive)
 - Неверно распознал как написанные AI 9% текстов из написанных человеком (false positive)

Поисковые машины на основе ИИ

- PHIND
 - <https://phind.com/>
- You.com
 - <https://you.com/>
- Perplexity
 - <https://www.perplexity.ai/>

Ограничения

- Иногда ответы ChatGPT выглядят достоверными, но на самом деле ошибочны
 - В процессе обучения моделей никак не оптимизируется правильность информации
- Модель не может обращаться к актуальным знаниям (БД, интернет)
 - Где находится Мисти парк в Кирове?
 - Какой код у буквы Я?
- Модель чувствительна к перефразировке вопроса
- Модель часто излишне многословна и использует клише
 - «Я – языковая модель, обученная OpenAI»
 - Инструкторы предпочитали длинные ответы
- В случае непонятных вопросов модель обычно предполагает, что имел в виду пользователь, а не переспрашивает
- Иногда не распознаются неприемлемые запросы
- Для русского языка – ошибки в грамматике

Выводы

- Причины успеха ChatGPT:
 - доступность
 - удобный интерфейс
 - усиленная безопасность
 - возможность уточнять (“Instruct”)
- Рецепт создания чат-бота:
 - Large Language Model
 - Fine-Tuning
 - Reinforcement Learning from Human Feedback (RLHF)
 - Rules
 - Retrieval augmented
- Ограничения (фактчекинг)
- Меняет ландшафт поиска

Ссылки: ChatGPT

- ChatGPT (OpenAI)
 - <https://openai.com/blog/chatgpt/>
- InstructGPT (OpenAI)
 - <https://openai.com/blog/instruction-following/>
- Игорь Котенков – ChatGPT как инструмент для поиска: решаем основную проблему (WebGPT)
 - <https://habr.com/ru/company/ods/blog/709222/>
- Alan Thompson – GPT-3.5 + ChatGPT: An illustrated overview
 - <https://lifearchitected.ai/chatgpt/>
- Awesome ChatGPT
 - <https://github.com/saharmor/awesome-chatgpt>
- Hugging Face – What Makes a Dialog Agent Useful?
 - <https://huggingface.co/blog/dialog-agents>

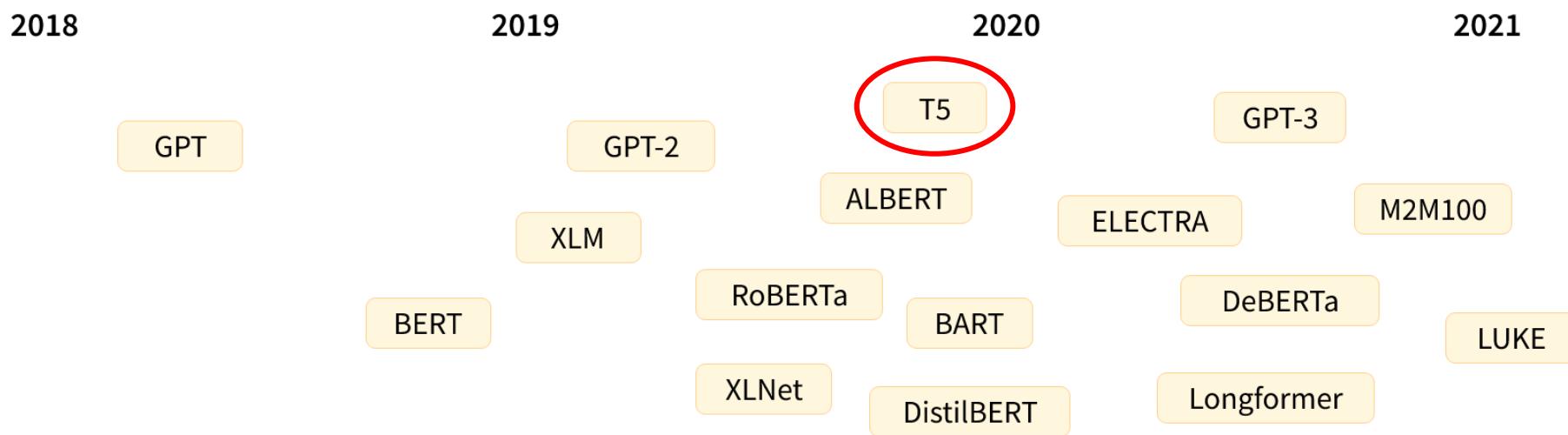
Ссылки: Reinforcement Learning with Human Feedback and Proximal Policy Optimization

- Andrej Karpathy – Deep Reinforcement Learning: Pong from Pixels
 - <http://karpathy.github.io/2016/05/31/rl/>
 - Перевод: <https://habr.com/ru/post/439674/>
- ИТМО – Методы policy gradient и алгоритм асинхронного актора-критика
 - http://neerc.ifmo.ru/wiki/index.php?title=Методы_policy_gradient_и_алгоритм_асинхронного_актора-критика
- Sergey Levine – CS 294: Deep Reinforcement Learning
 - <http://rail.eecs.berkeley.edu/deeprlcourse-fa17/>
- OpenAI – Proximal Policy Optimization
 - <https://openai.com/blog/openai-baselines-ppo/>
- Schulman J. et al. Proximal Policy Optimization Algorithms
 - <https://arxiv.org/abs/1707.06347>
- Hugging Face – Illustrating Reinforcement Learning from Human Feedback (RLHF)
 - <https://huggingface.co/blog/rlhf>
- Hugging Face – Deep Reinforcement Learning Course
 - <https://huggingface.co/learn/deep-rl-course>

Модель T5

2024

Ключевые модели

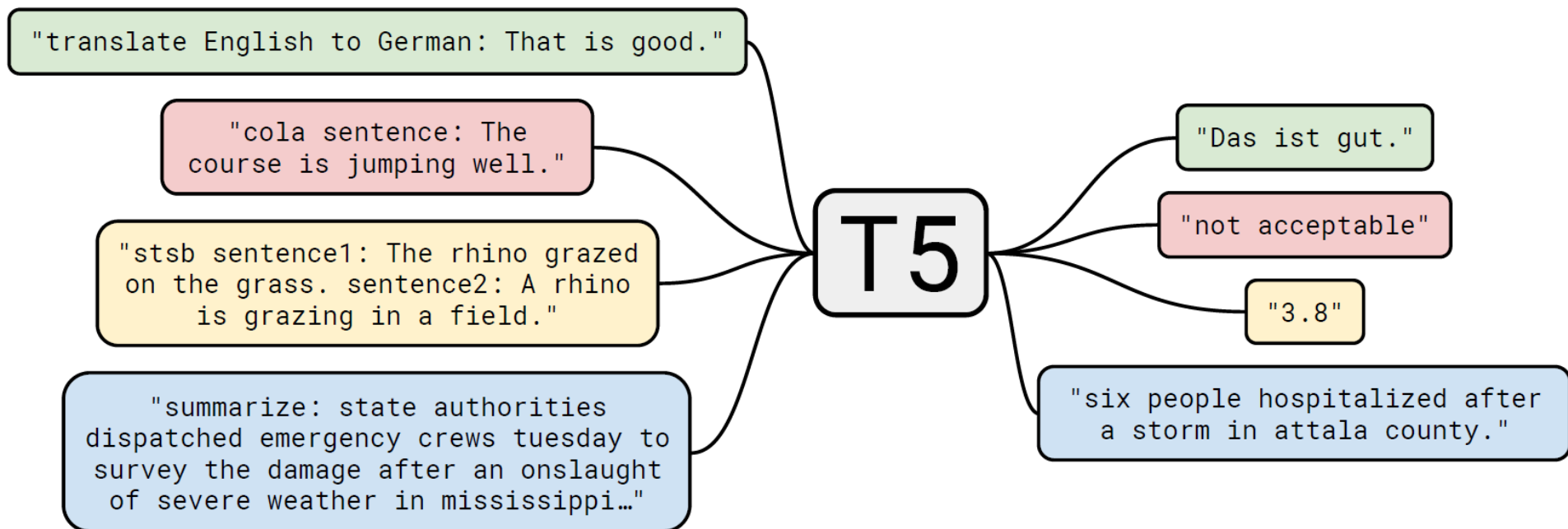


T5

- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer // Journal of Machine Learning Research. 2020. Vol. 21. P. 1–67
 - <https://jmlr.org/papers/volume21/20-074/20-074.pdf>
 - <https://arxiv.org/abs/1910.10683>
 - Google

Идея

- Основная идея – любая задача обрабатывается как **text-to-text**



Идея

- Основная идея – любая задача обрабатывается как **text-to-text**
 - T5 = Text-to-Text Transfer Transformer
- Такая постановка позволяет использовать одну и ту же модель, целевую функцию, процедуры обучения и декодирования для всех задач
 - Multi-task Learning

Вклад

- Обучают несколько моделей (до 11 млрд)
- Вводят корпус Colossal Clean Crawled Corpus (C4)
Колоссальный очищенный собранный в интернете корпус
- Получают SOTA результаты на нескольких задачах:
 - Классификация высказываний
 - Ответы на вопросы
 - Перевод и реферирование текста
- Код, данные и модели предоставляют в общий доступ
 - <https://github.com/google-research/text-to-text-transfer-transformer>

Модель

- Модель – Transformer
- Отличия от оригинального Transformer:
 - слои нормализации (T5LayerNorm) располагаются до, а не после слоя внутреннего внимания и полносвязного слоя
 - в слоях нормализации отсутствует смещение (bias)
 - используется относительное позиционное кодирование вместо синусоидального

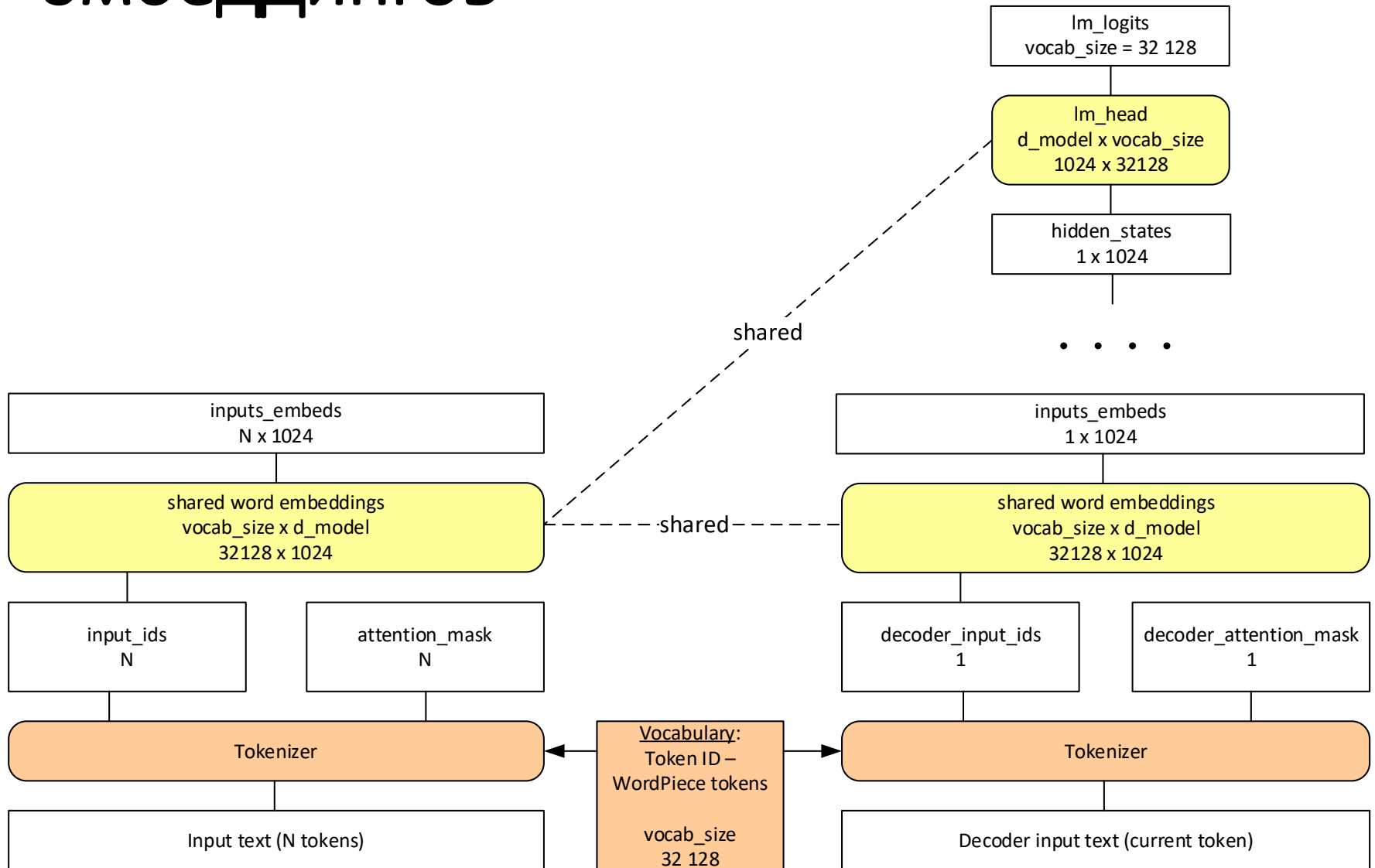
Токенизация

- Токенизация – WordPiece (как в BERT)
 - Максимизируется правдоподобие
 - Размер словаря: 32 128 токенов
 - «Посмотри в энциклопедическом словаре.» →
 - Посмотри
 - В
 - энциклопеди
 - ческом
 - слова
 - ре
 - .

Связывание входных и выходных эмбеддингов

- Веса входных эмбеддингов (как энкодера, так и декодера) совпадают с весами на последнем слое, который преобразует скрытые вектора на выходе модели в вероятности генерируемых токенов – это один и тот же слой
- Такой прием называется *связывание входных и выходных эмбеддингов* (embedding weight tying) или *объединенные входные и выходные эмбеддинги* (joint input-output embeddings):
 - [Press O., Wolf L. Using the Output Embedding to Improve Language Models \(2016\)](#)
 - [Roth T. How does weight-tying work in a RNN? \(2020\)](#)
 - [Papers with code. Weight Tying](#)

Связывание входных и выходных эмбеддингов

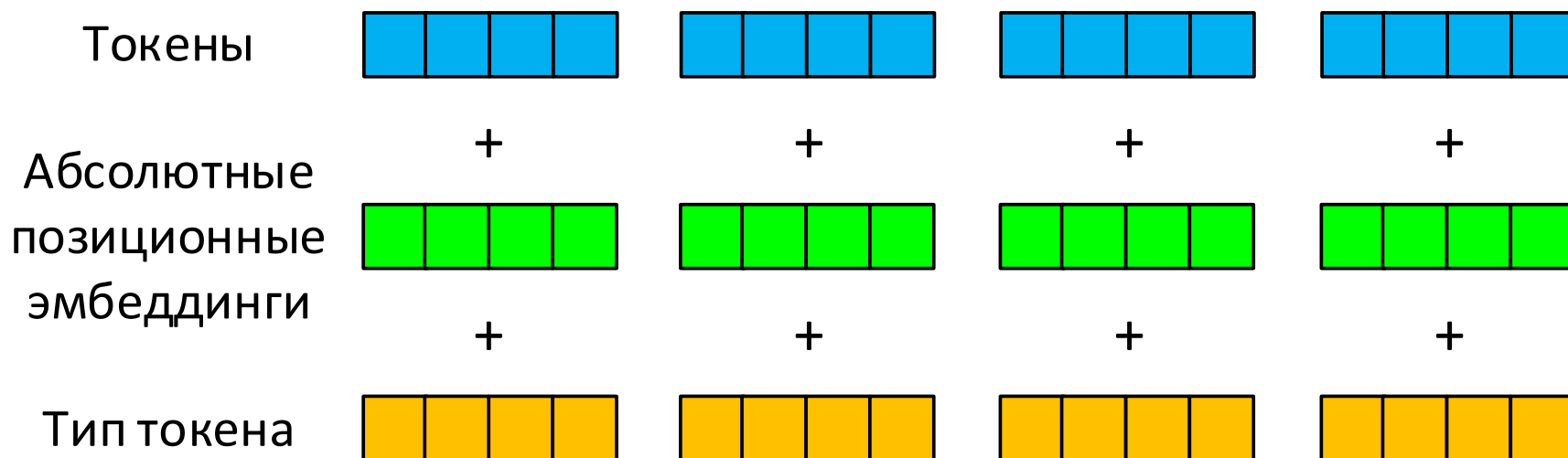


Относительное позиционное кодирование

- Для указания взаимного расположения токенов используется относительное позиционное кодирование [Shaw et al., 2018]
- Вместо того, чтобы задавать для каждой входной позиции фиксированные эмбединги, вводятся относительные эмбединги, соответствующие расстоянию между «ключом» и «запросом» в механизме внутреннего внимания
- Расстояния «влево» и «вправо» от токена различаются
- Эти эмбединги настраиваются в процессе обучения

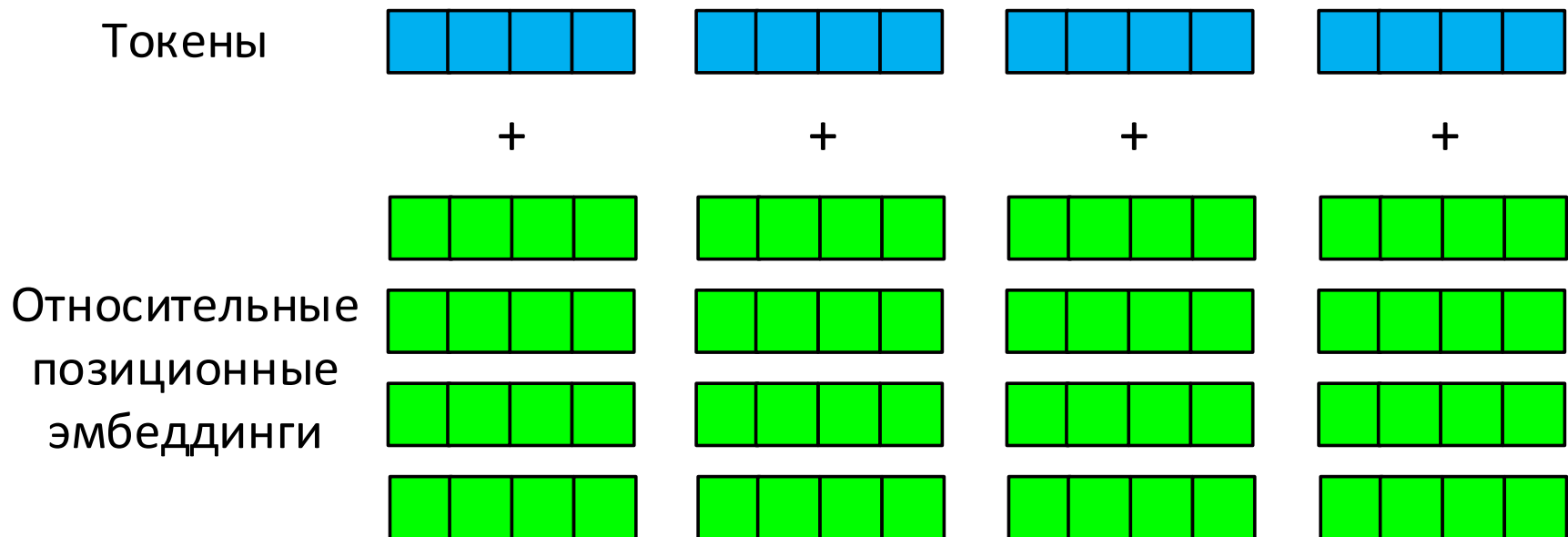
Относительное позиционное кодирование

BERT

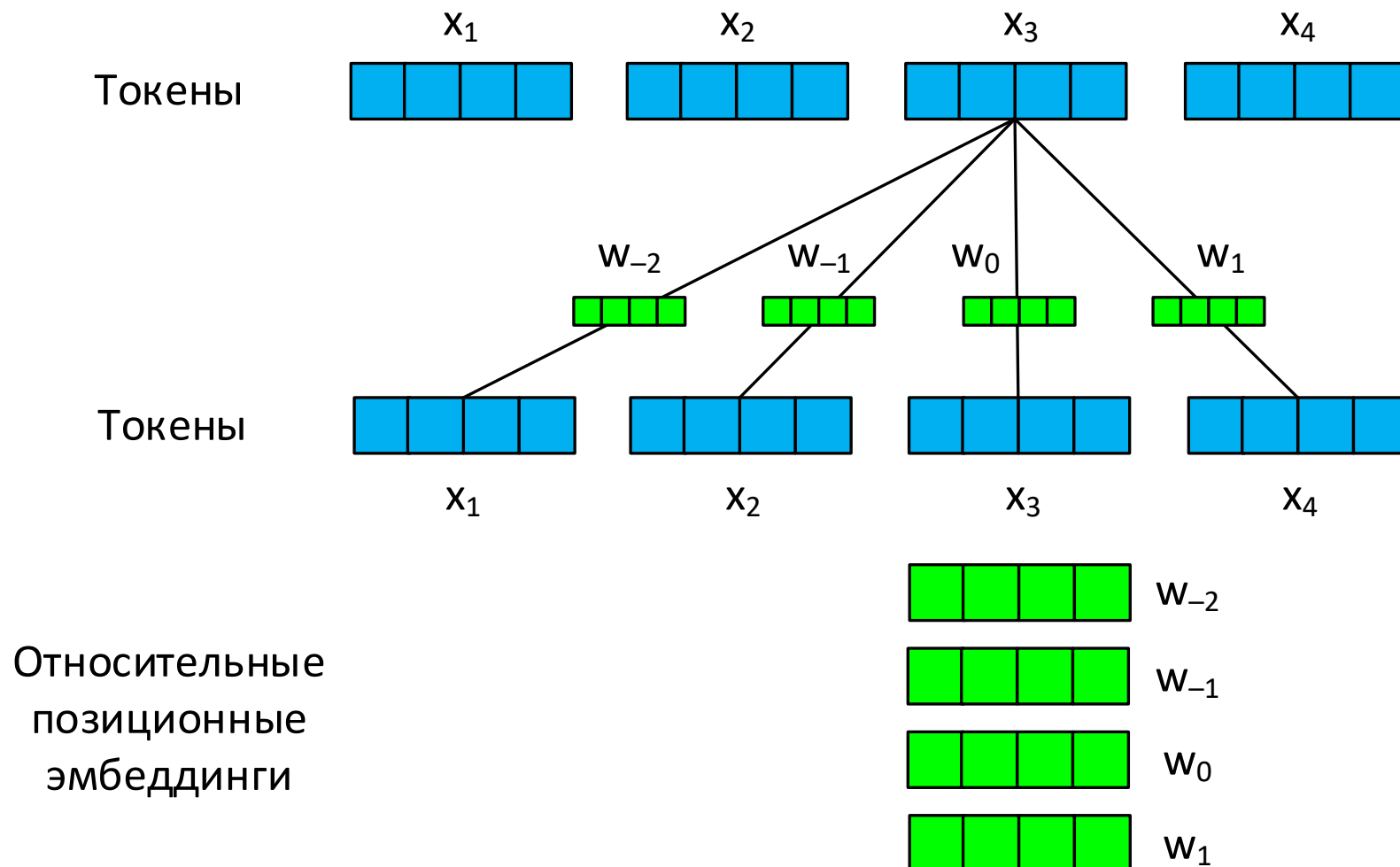


Относительное позиционное кодирование

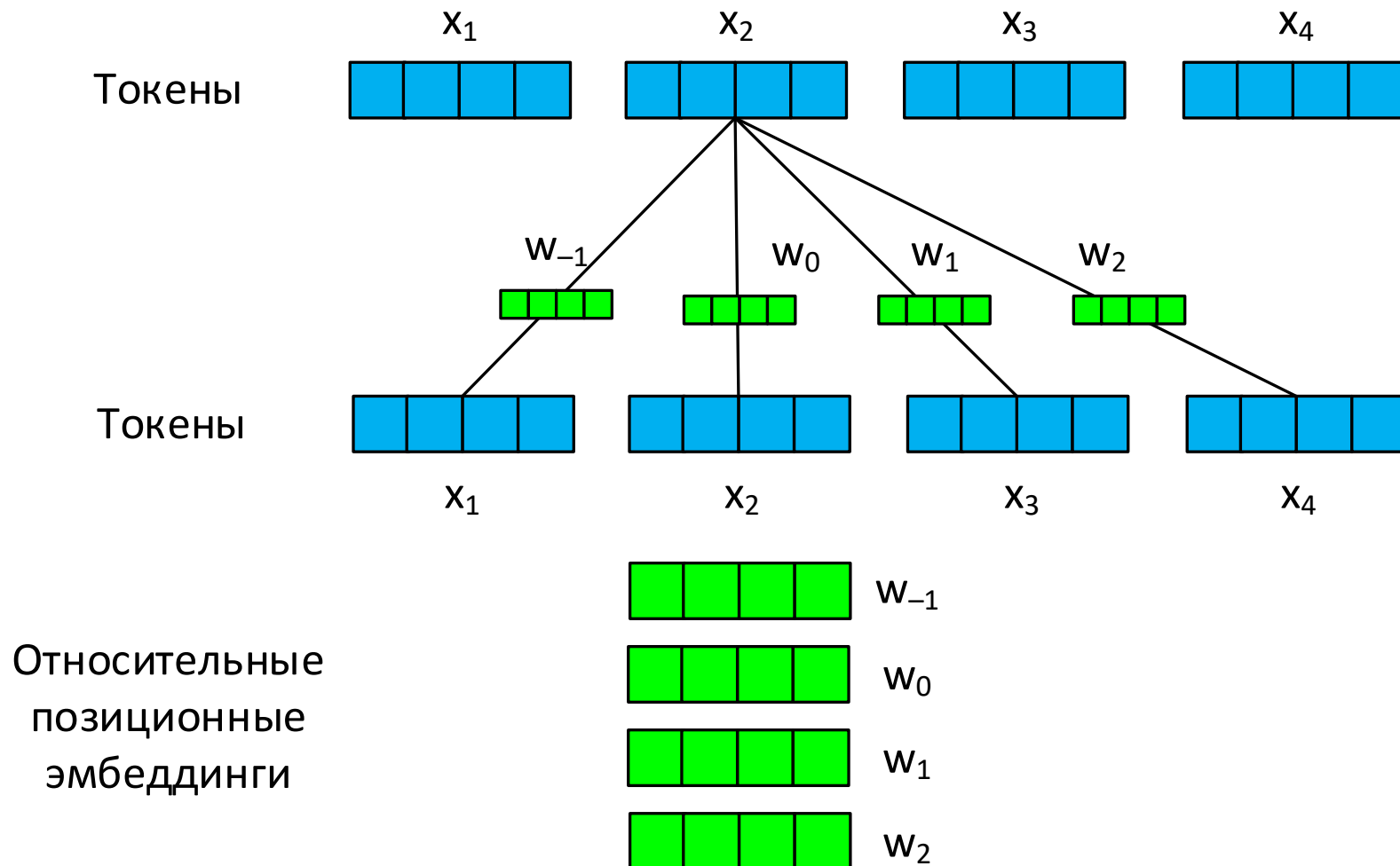
Shaw, Uszkoreit, Vaswani – Self-Attention
with Relative Position Representations (2018)



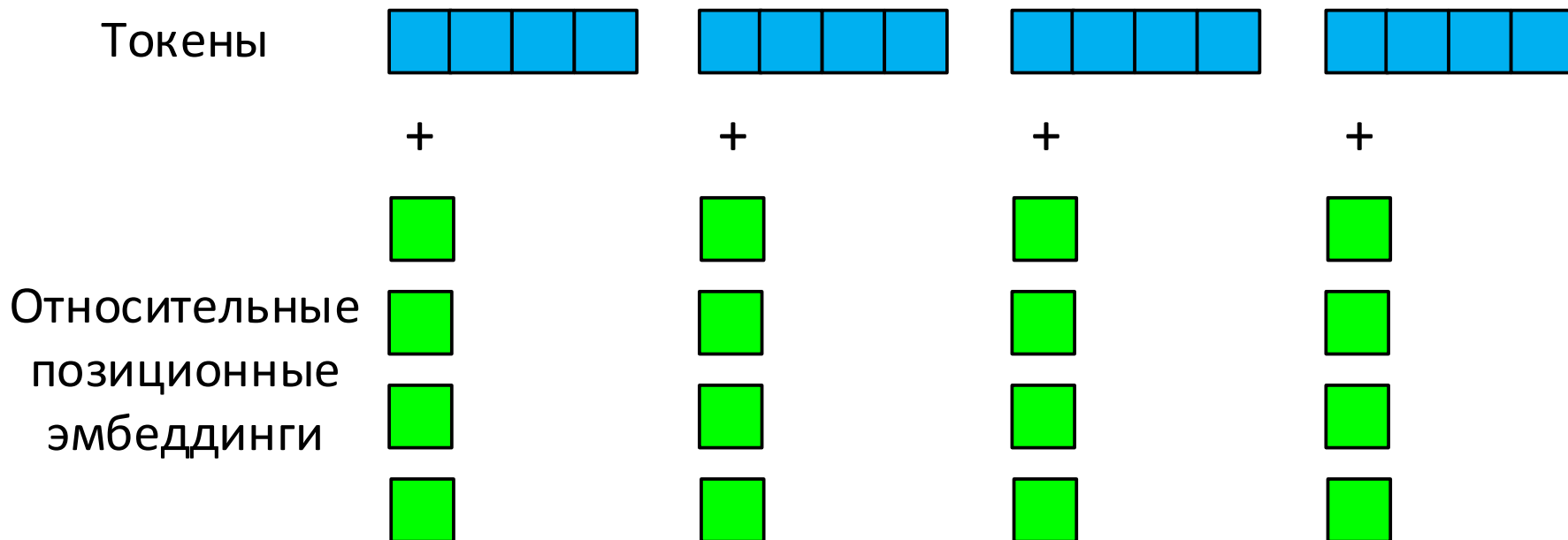
Относительное позиционное кодирование



Относительное позиционное кодирование

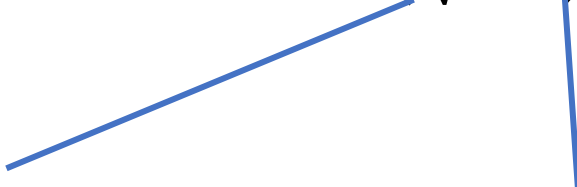


Относительное позиционное кодирование



Относительное позиционное кодирование

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

$$e_{ij} = \frac{1}{\sqrt{d}} (x_i W^Q)(x_j W^K) + \mathbf{b}_{j-i}$$
Two blue lines originate from the denominator $\sqrt{d_k}$ in the first equation. One line extends downwards and to the left, pointing towards the denominator $\sqrt{d}$ in the second equation. The other line extends downwards and to the right, pointing towards the bias term $\mathbf{b}_{j-i}$ in the second equation.

Относительное позиционное кодирование

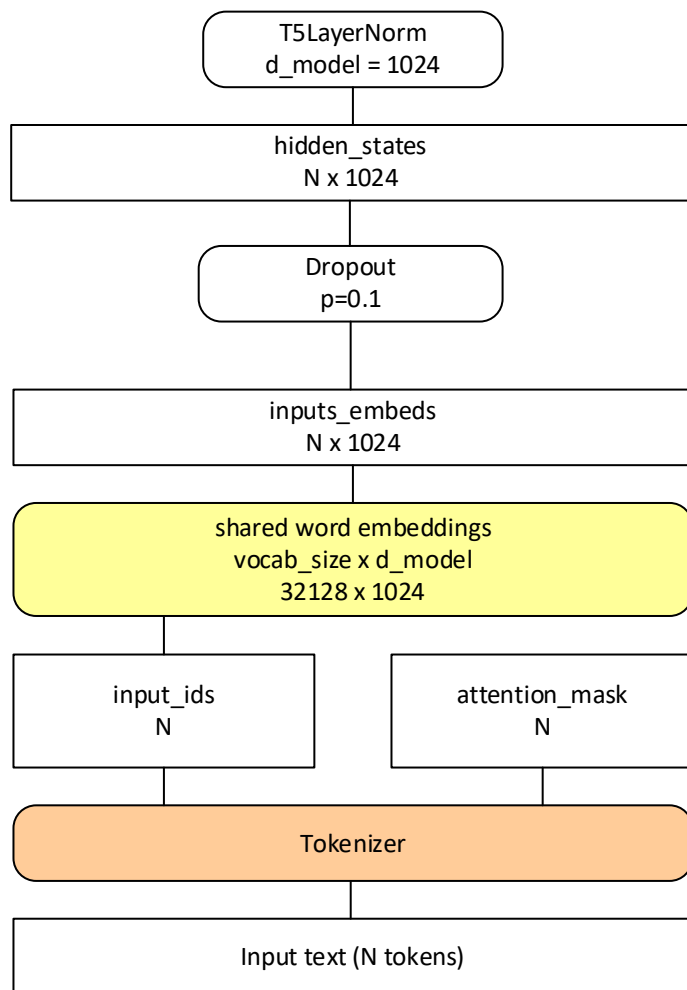
- Максимально допустимое расстояние является параметром: `relative_attention_max_distance=128`),
- Все расстояния, большие 128 , будут представляться в виде одного и того же эмбединга
 - Расстояния, меньшие -128, так же будут представляться в виде одного и того же эмбединга

Относительное позиционное кодирование

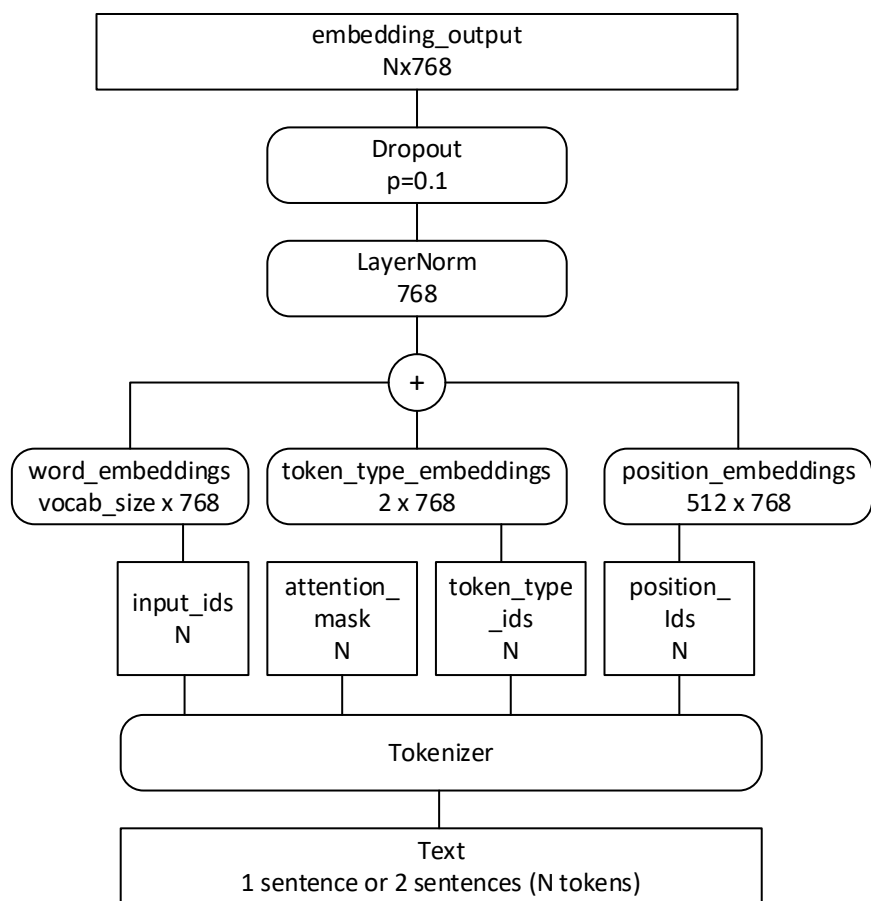
- Теоретически, размер входа T5 ограничен только памятью, но:
 - требования к памяти растут квадратично от размера входа
 - нужно учитывать, на каких последовательностях модель обучалась (как правило, не более 512 токенов)

Вход модели

T5

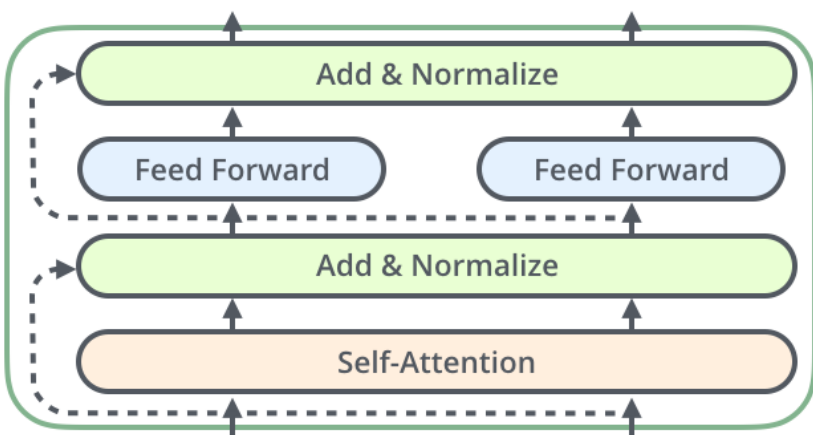


BERT

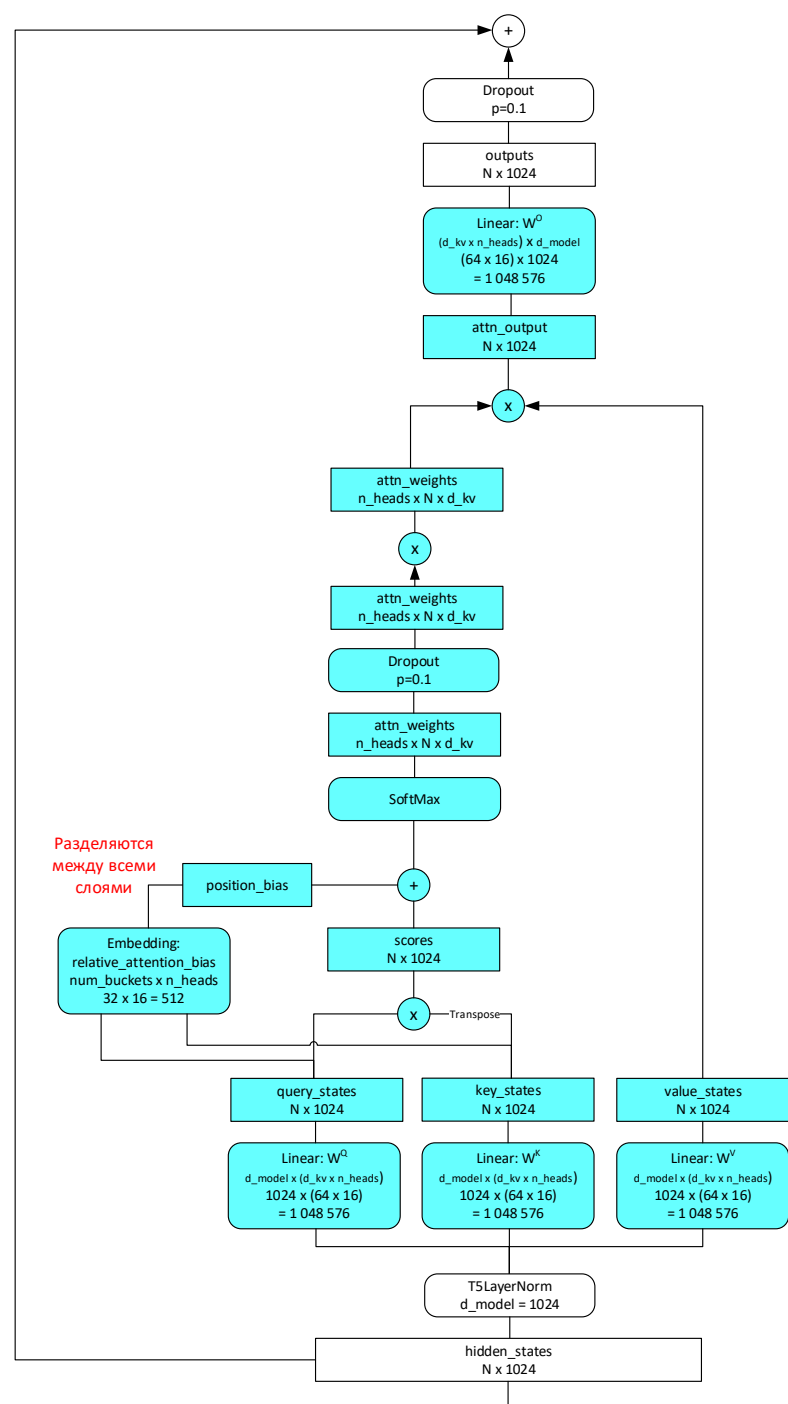


Self-Attention

T5LayerSelfAttention

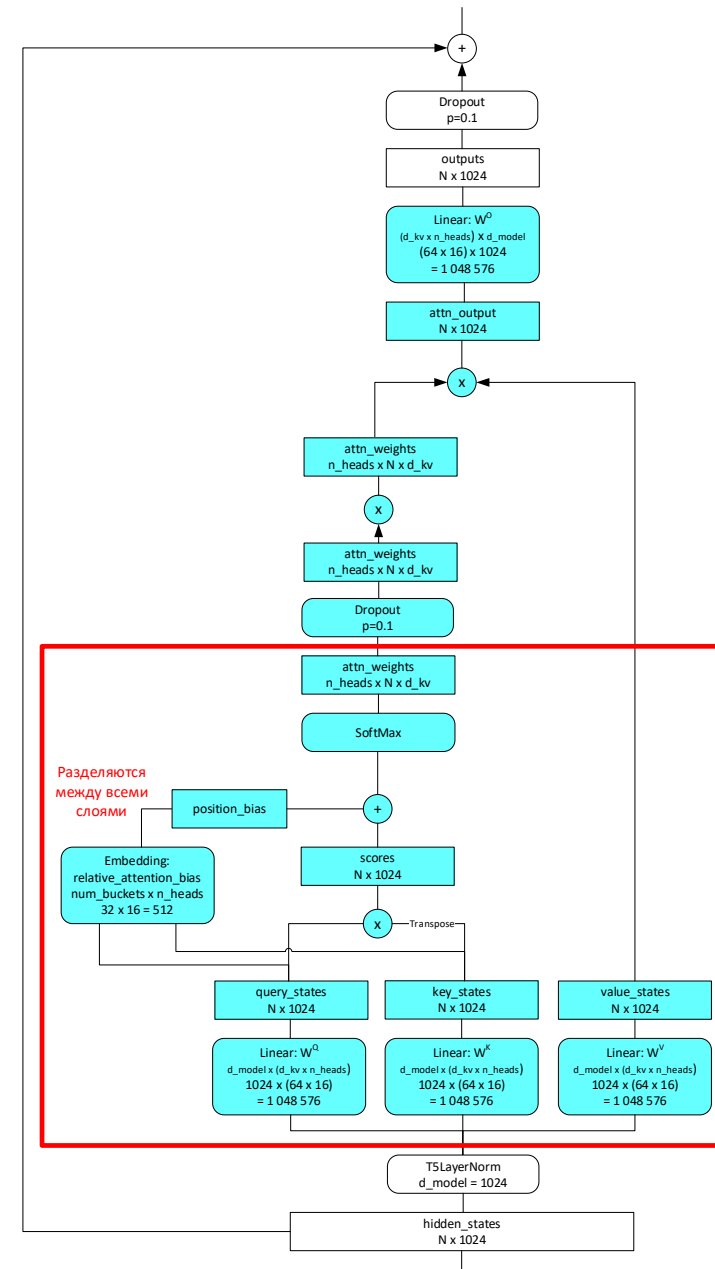
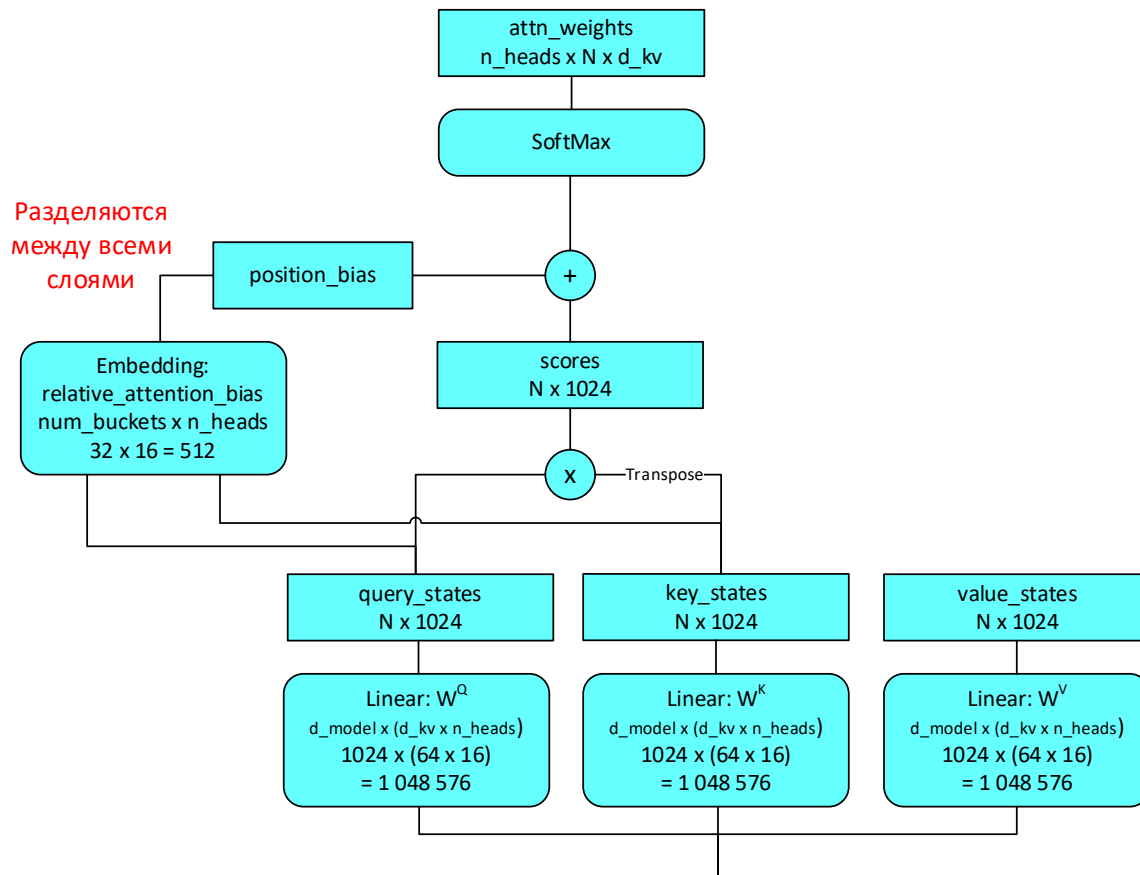


T5Attention



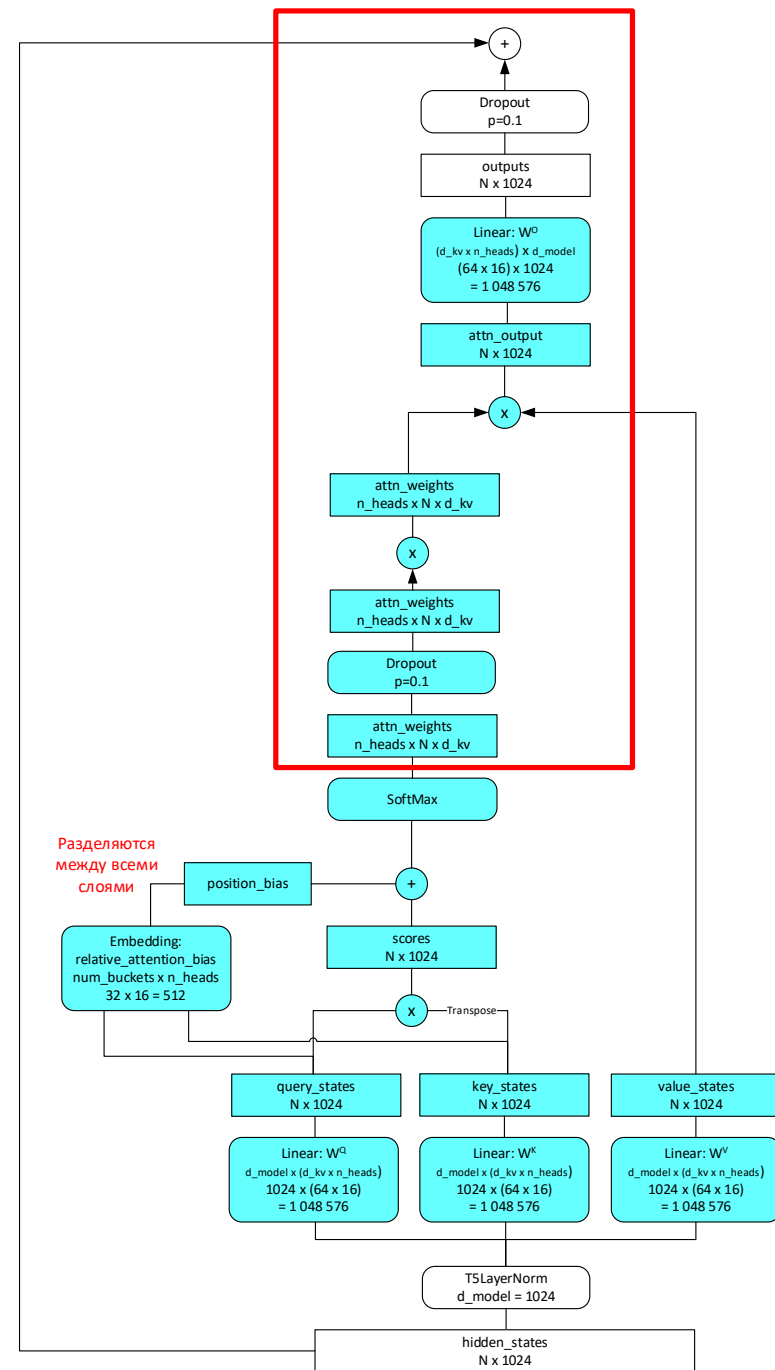
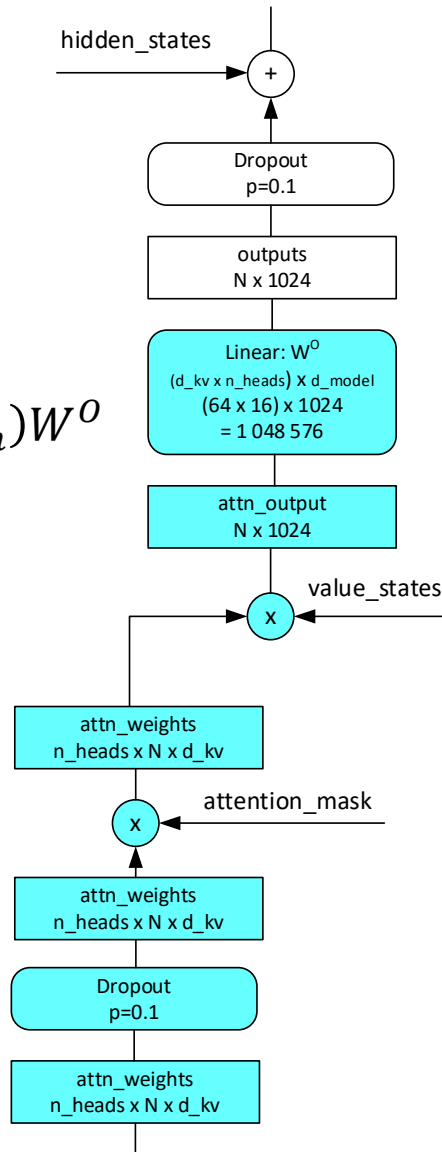
Self-Attention

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



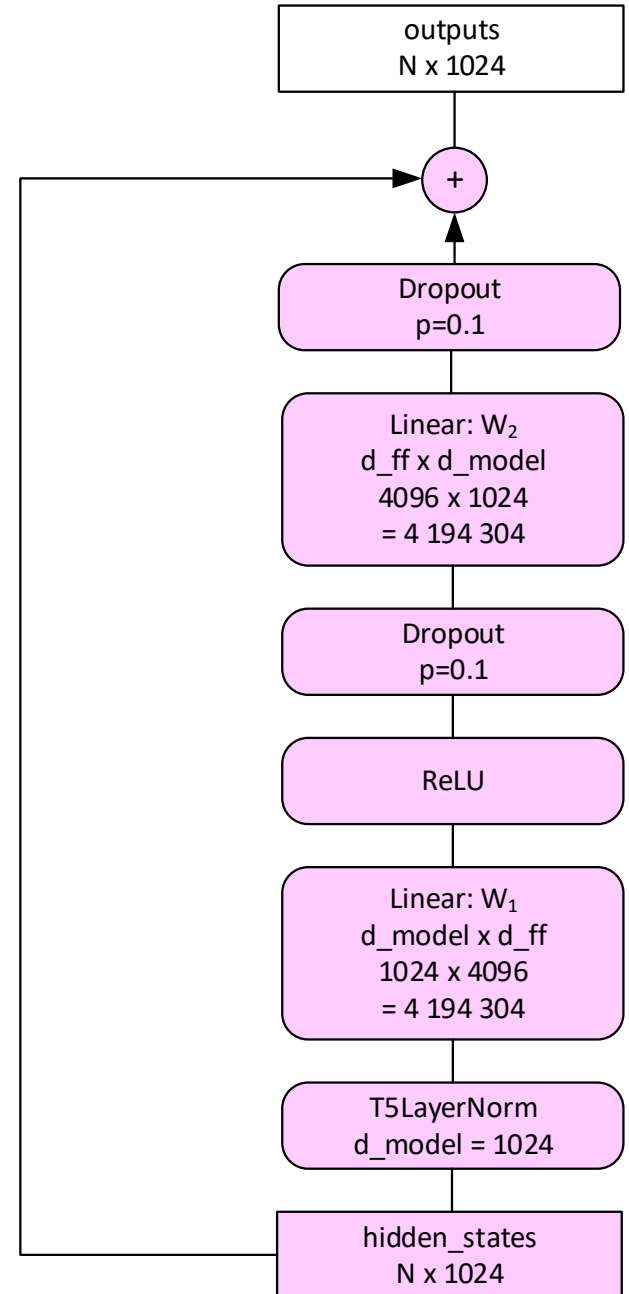
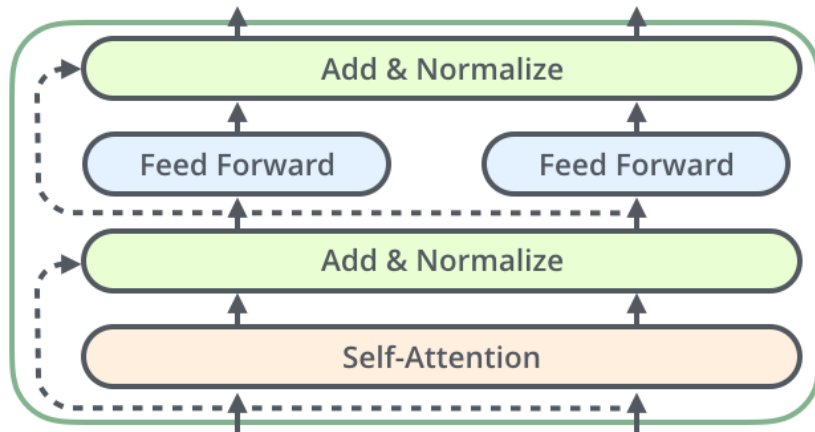
Self-Attention

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O$$

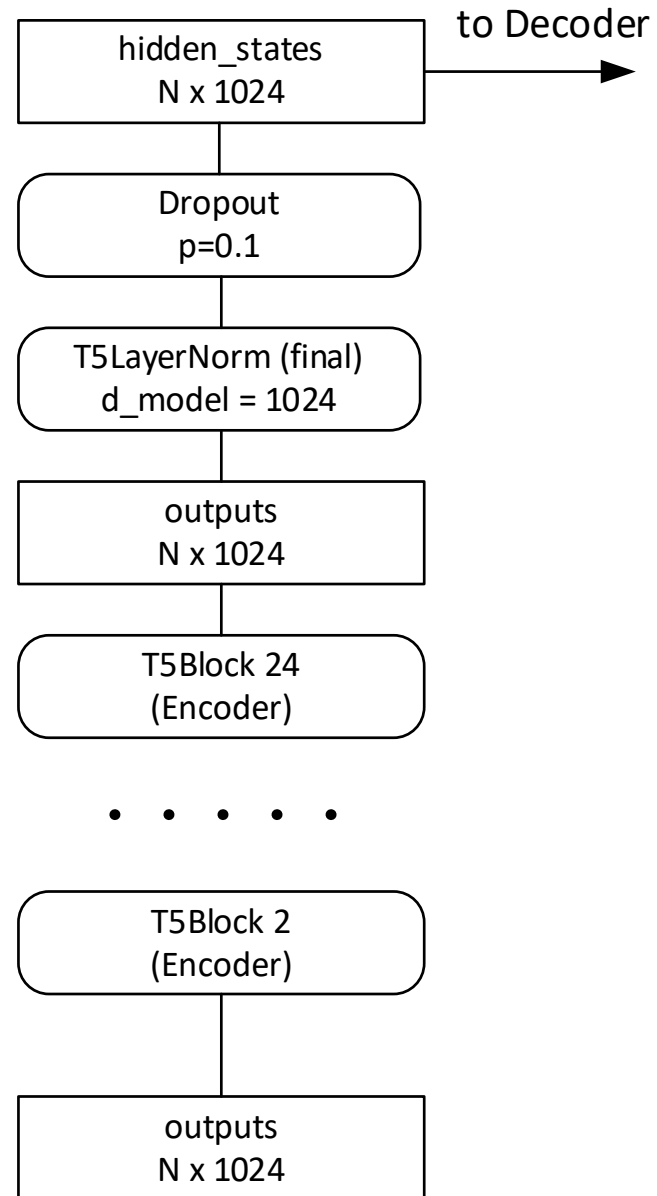


Feed-Forward Neural Network

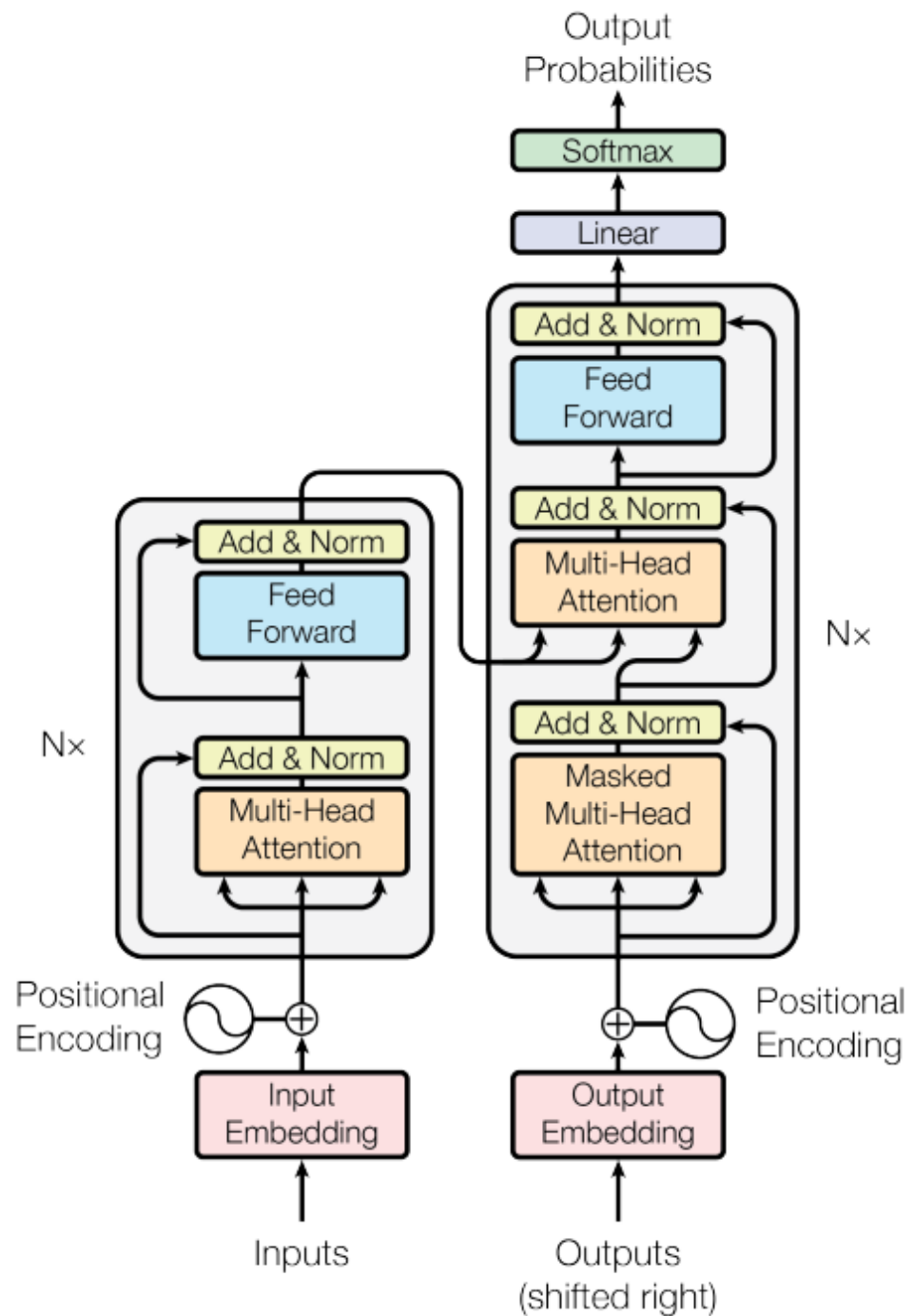
T5LayerFF



Encoder

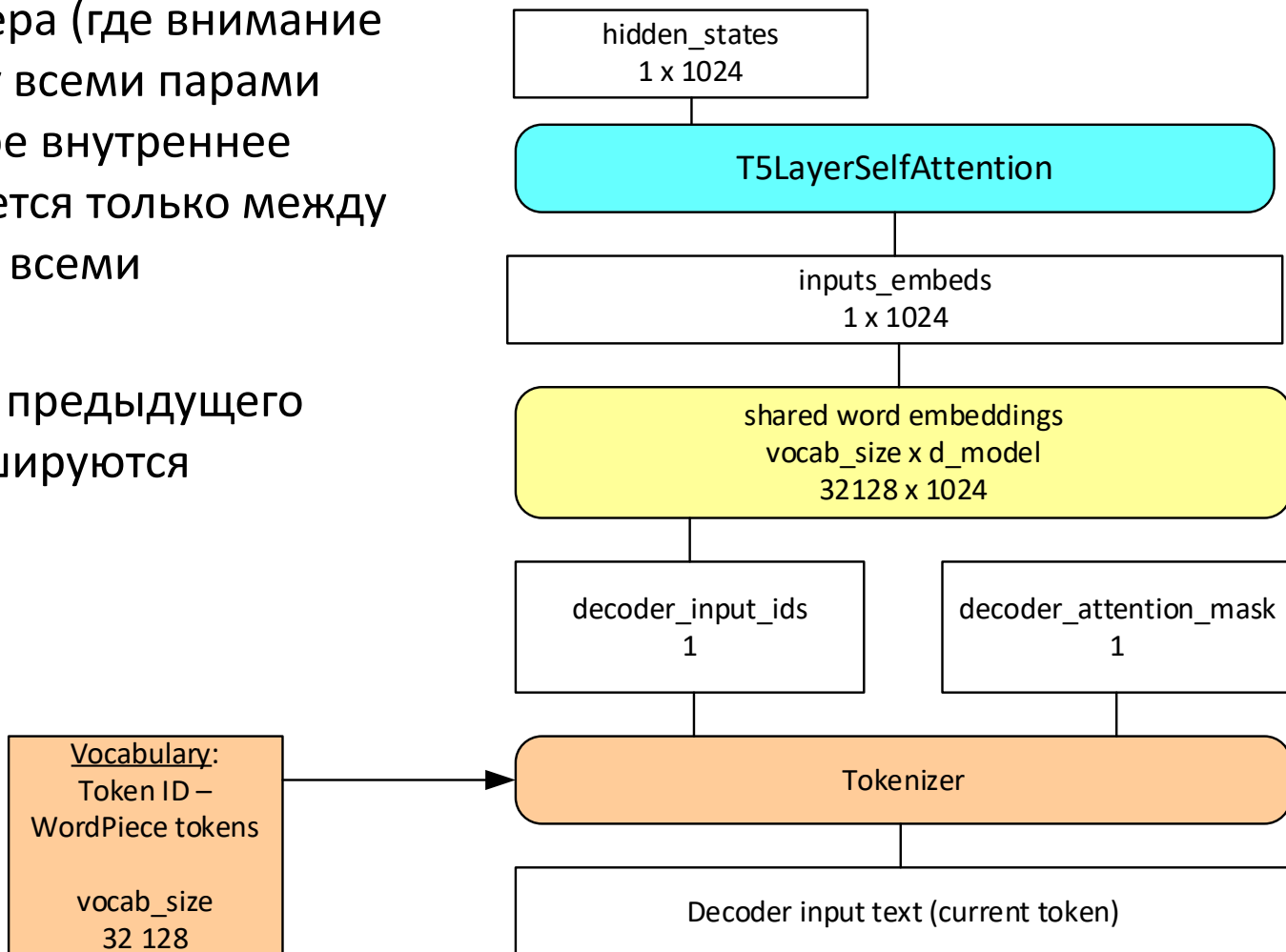


Decoder

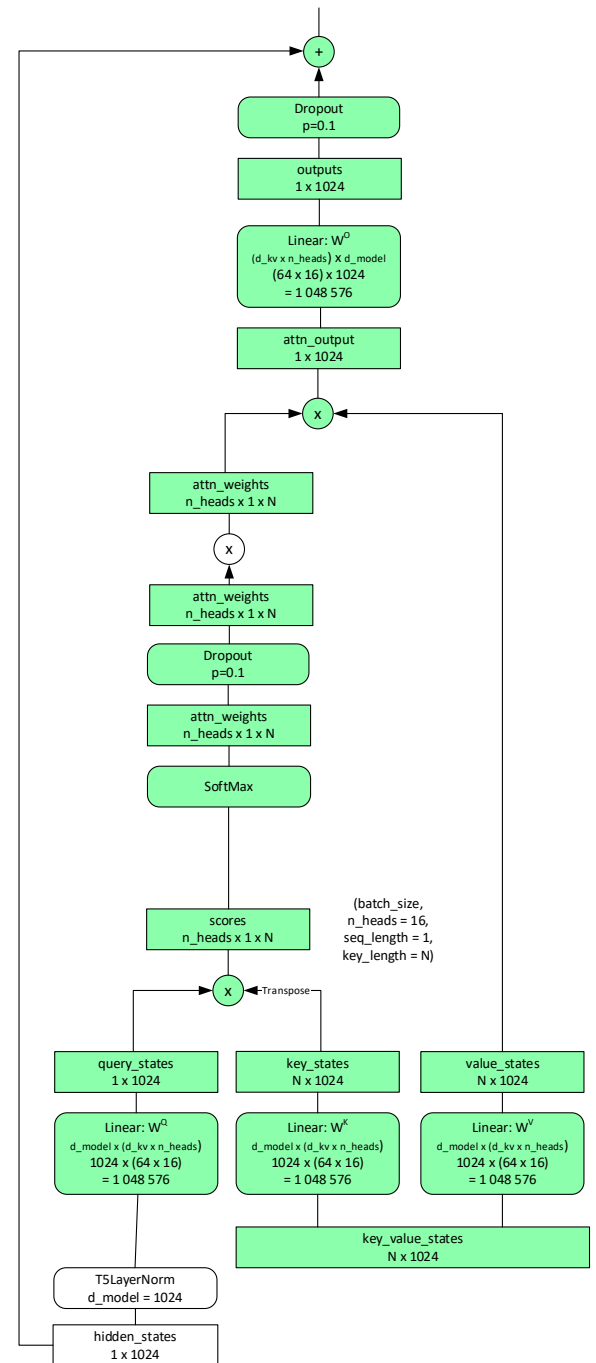


Decoder – вход

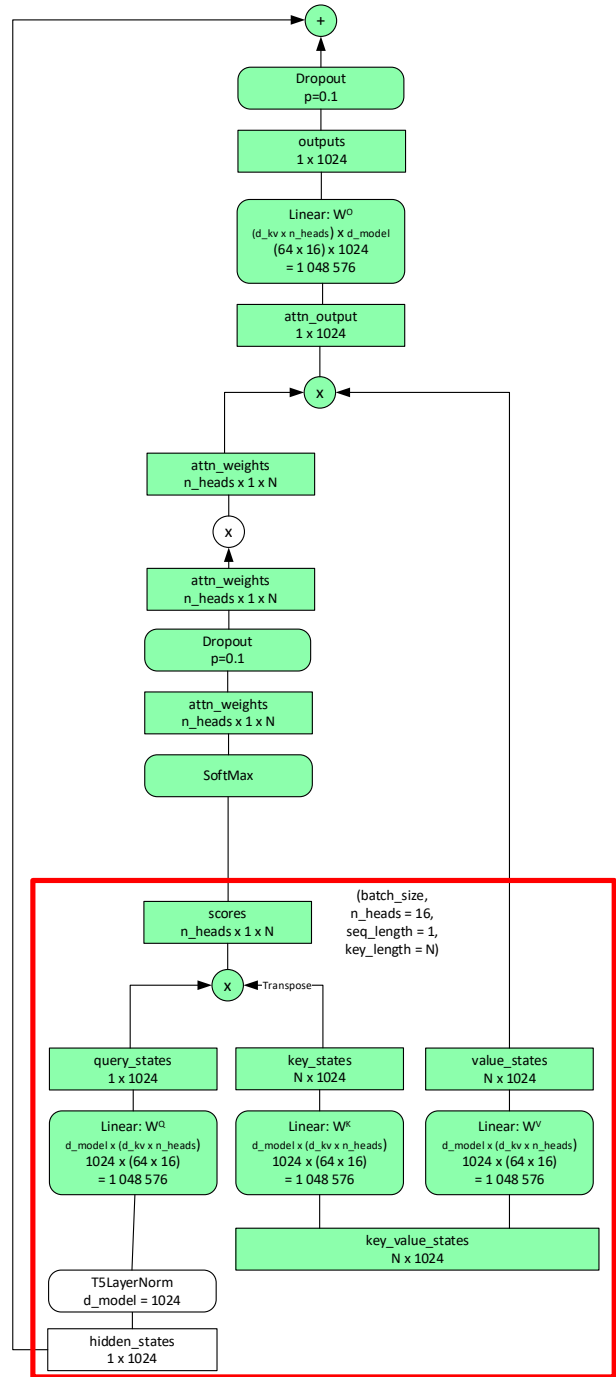
- В отличие от энкодера (где внимание вычисляется между всеми парами токенов), в декодере внутреннее внимание вычисляется только между текущим токеном и всеми предыдущими
- Все keys и values с предыдущего шага генерации кэшируются (past_key_value)



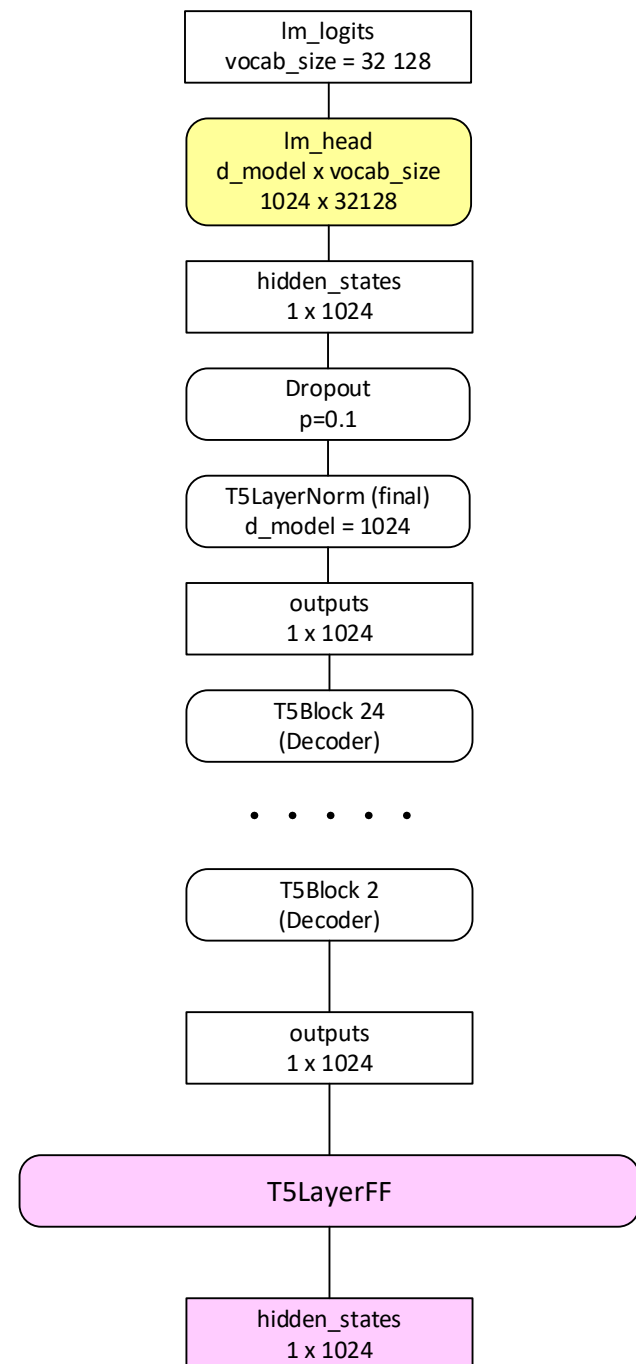
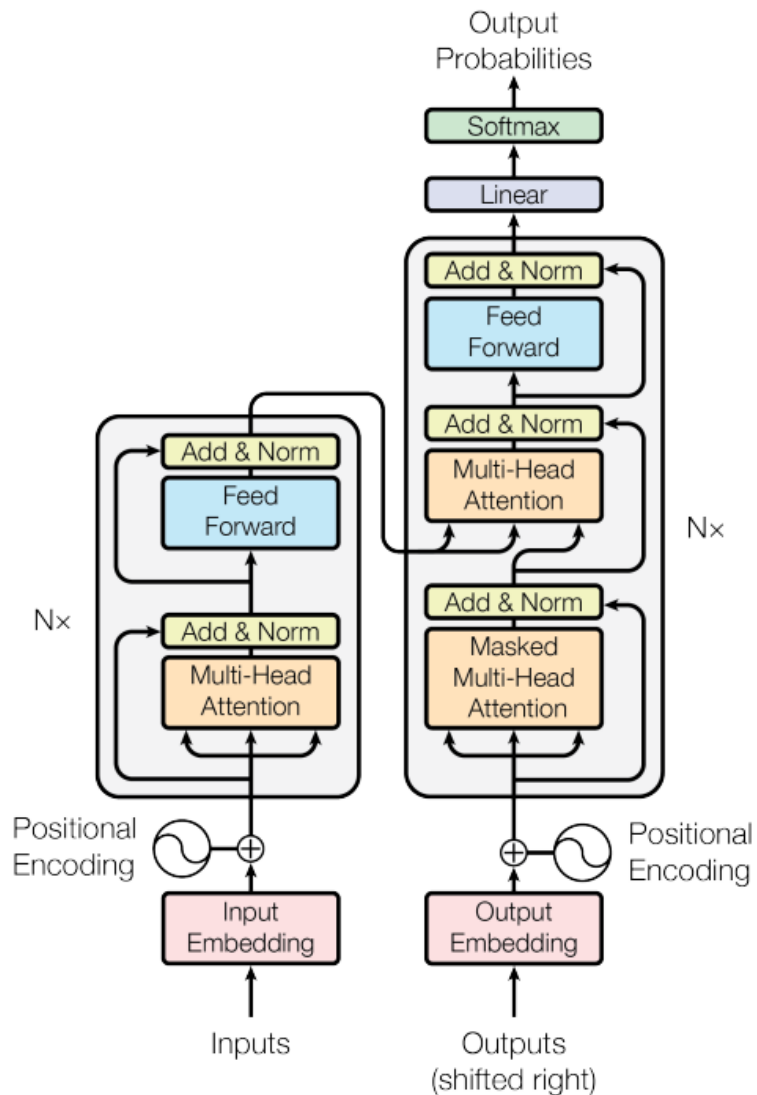
T5LayerCrossAttention



T5LayerCrossAttention



Decoder



Common Crawl

- <https://commoncrawl.org>
- 20 ТБ в месяц
- Апрель 2019: очистка, английский язык → 750 ГБ
- Colossal Clean Crawled Corpus (C4)
 - <https://www.tensorflow.org/datasets/catalog/c4>
 - 365 млн документов (807 ГБ)

Unsupervised objective

- Задача восстановления входа (denoising autoencoder)
- Похоже на BERT, но маскируются последовательности токенов (в среднем 3) – 15% от входа
- Используются служебные токены <extra_id_N>

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>

Supervised objectives

- Sentence acceptability judgment (CoLA)
- Sentiment analysis (SST-2)
- Paraphrasing/sentence similarity (MRPC, STS-B, QQP)
- Natural language inference (MNLI, QNLI, RTE, CB)
- Coreference resolution (WNLI, WSC)
- Sentence completion (COPA)
- Word sense disambiguation (WIC)
- Question answering (MultiRC, ReCoRD, BoolQ, SQuAD)
- Abstractive summarization (CNN/Daily Mail)
- Translation (WMT English to German, French, Romanian)

Задачи

Перевод:

- **Вход:** **translate English to French:** This image section from an infrared recording by the Spitzer telescope shows a "family portrait" of countless generations of stars: the oldest stars are seen as blue dots, while more difficult to identify are the pink-coloured "new-borns" in the star delivery room.
- **Выход:** Ce détail d'une photographie infrarouge prise par le télescope Spitzer montre un "portrait de famille" des innombrables générations d'étoiles: les plus vieilles étoiles sont en bleu et les points roses, plus difficiles à identifier, sont les "nouveau-nés" dans la salle d'accouchement de l'univers.

Задачи

Тональность (SST – Stanford Sentiment Treebank):

- **Вход:** **sst2 sentence**: it confirms fincher 's status as a film maker who artfully bends technical know-how to the service of psychological insight
- **Выход:** positive

Логичность (MNLI – Multi-Genre Natural Language Inference):

- **Вход:** **mnli hypothesis**: My feelings towards pigeons are filled with animosity. **premise**: I hate pigeons
- **Выход:** entailment

Задачи

Ответы на вопросы (SQuAD – Stanford Question Answering Dataset):

- **Вход:** **question:** What does increased oxygen concentrations in the patient's lungs displace? **context:** Hyperbaric (high-pressure) medicine uses special oxygen chambers to increase the partial pressure of O₂ around the patient and, when needed, the medical staff. Carbon monoxide poisoning, gas gangrene, and decompression sickness (the 'bends') are sometimes treated using these devices. Increased O₂ concentration in the lungs helps to displace carbon monoxide from the heme group of hemoglobin. Oxygen gas is poisonous to the anaerobic bacteria that cause gas gangrene, so increasing its partial pressure helps kill them. Decompression sickness occurs in divers who decompress too quickly after a dive, resulting in bubbles of inert gas, mostly nitrogen and helium, forming in their blood. Increasing the pressure of O₂ as soon as possible is part of the treatment.
- **Выход:** carbon monoxide

Задачи

Семантическое сходство (STSB – Semantic Textual Similarity Benchmark):

- **Вход:** **stsb sentence1**: There must be a balance as a whole.
sentence2: The body must be in balance.
- **Выход:** 4.0
 - Регрессия → 21-классовая классификация

Обучение

- Unsupervised pre-training:
 - 524 288 шагов на C4
 - AdaFactor
 - Батч: 128 последовательностей по 512 токенов (65K токенов)
 - 34 млрд токенов всего в процессе обучения (не весь C4)
 - BERT: 137 млрд токенов
 - RoBERTa: 2.2 трлн токенов
- Supervised fine-tuning:
 - 262 144 шага
 - Тот же размер батча

Модели

Англоязычные модели

(supervised и unsupervised tasks для английского языка):

- [t5-small](#) – 60 млн параметров
- [t5-base](#) – 222 млн параметров
- [t5-large](#) – 737 млн параметров
- [t5-3b](#) – 3 млрд параметров
- [t5-11b](#) - 11 млрд параметров

Версия 1.1 (только unsupervised pre-training): [ссылка](#)

Модели

Многоязычные модели

(unsupervised task для 101 языка: русский – на втором месте по количеству обучающих данных):

- [mt5-small](#) – 300 млн параметров
- [mt5-base](#) – 580 млн параметров
- [mt5-large](#) – 1.2 млрд параметров
- [mt5-xl](#) – 3.7 млрд параметров
- [mt5-xxl](#) – 13 млрд параметров

Модели

Русскоязычные модели (Сбер; только unsupervised pre-training)

- ruT5-base ([HuggingFace](#), [SberCloud](#)) – 222 млн параметров
- ruT5-large ([HuggingFace](#), [SberCloud](#)) – 737 млн параметров

Результаты: GLUE

Model	GLUE Average
Previous best	89.4 ^a
T5-Small	77.4
T5-Base	82.7
T5-Large	86.4
T5-3B	88.5
T5-11B	90.3

[Ссылка](#)

Rank	Name	Model	URL	Score
1	Microsoft Alexander v-team	Turing ULR v6		91.3
2	JDExplore d-team	Vega v1		91.3
3	Microsoft Alexander v-team	Turing NLR v5		91.2
4	DIRL Team	DeBERTa + CLEVER		91.1
5	ERNIE Team - Baidu	ERNIE		91.1
6	AliceMind & DIRL	StructBERT + CLEVER		91.0
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8
8	HFL iFLYTEK	MacALBERT + DKM		90.7
9	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6
10	T5 Team - Google	T5		90.3

Результаты: SuperGLUE

	Rank	Name	Model	URL	Score
	1	JDEExplore d-team	Vega v2		91.3
+	2	Liam Fedus	ST-MoE-32B		91.2
	3	Microsoft Alexander v-team	Turing NLR v5		90.9
	4	ERNIE Team - Baidu	ERNIE 3.0		90.6
	5	Yi Tay	PaLM 540B		90.4
+	6	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4
+	7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
	8	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8
+	9	T5 Team - Google	T5		89.3
	10	SPoT Team - Google	Frozen T5 1.1 + SPoT		89.2

[Ссылка](#)

Результаты: Russian SuperGLUE

Rank	Name	Team	Link	Score
1	HUMAN BENCHMARK	AGI NLP	i	0.811
2	Mistral 7B LoRA	Saiga team	i	0.763
3	FRED-T5 1.7B finetune	SberDevices	i	0.762
4	Golden Transformer v2.0	Avengers Ensemble	i	0.755
5	LLaMA-2 13B LoRA	Saiga team	i	0.718
6	Saiga 13B LoRA	Saiga team	i	0.712
7	YaLM p-tune (3.3B frozen + 40k trainable params)	Yandex	i	0.711
8	FRED-T5 large finetune	SberDevices	i	0.706
9	RuLeanALBERT	Yandex Research	i	0.698
10	FRED-T5 1.7B (only encoder 760M) finetune	SberDevices	i	0.694
11	ruT5-large finetune	SberDevices	i	0.686
12	ruRoberta-large finetune	SberDevices	i	0.684
13	gpt-3.5-turbo zero-shot	Saiga team	i	0.682
14	Golden Transformer v1.0	Avengers Ensemble	i	0.679
15	xlm-roberta-large (Facebook) finetune	SberDevices	i	0.654
16	mdeberta-v3-base (Microsoft) finetune	SberDevices	i	0.651
17	Saiga2 70B zero-shot	Saiga team	i	0.643
18	Saiga Mistral 7B zero-shot	Saiga team	i	0.635
19	ruT5-base finetune	Sberdevices	i	0.635
20	ruBert-large finetune	SberDevices	i	0.62

[Ссылка](#)

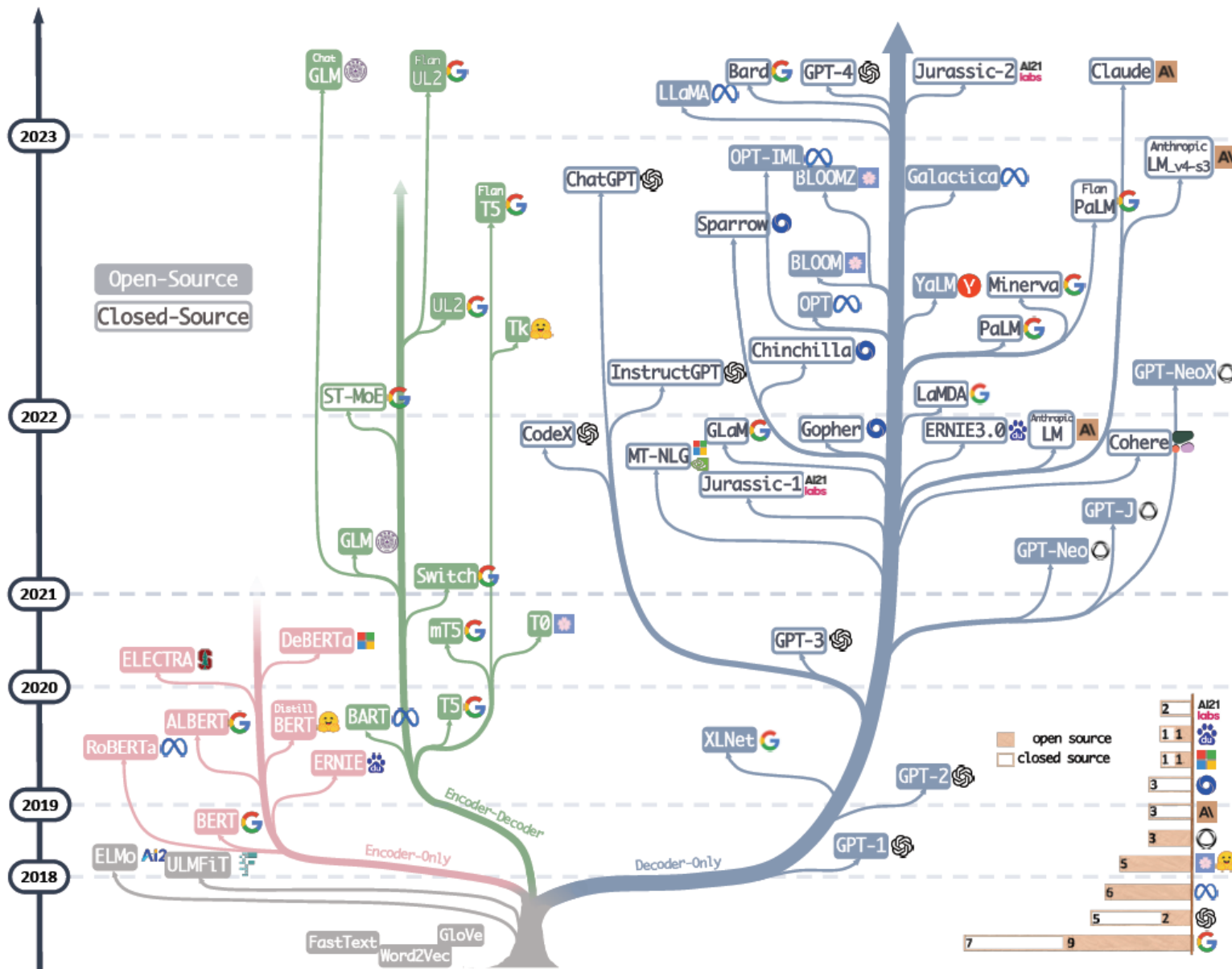
Обзор языковых моделей

План

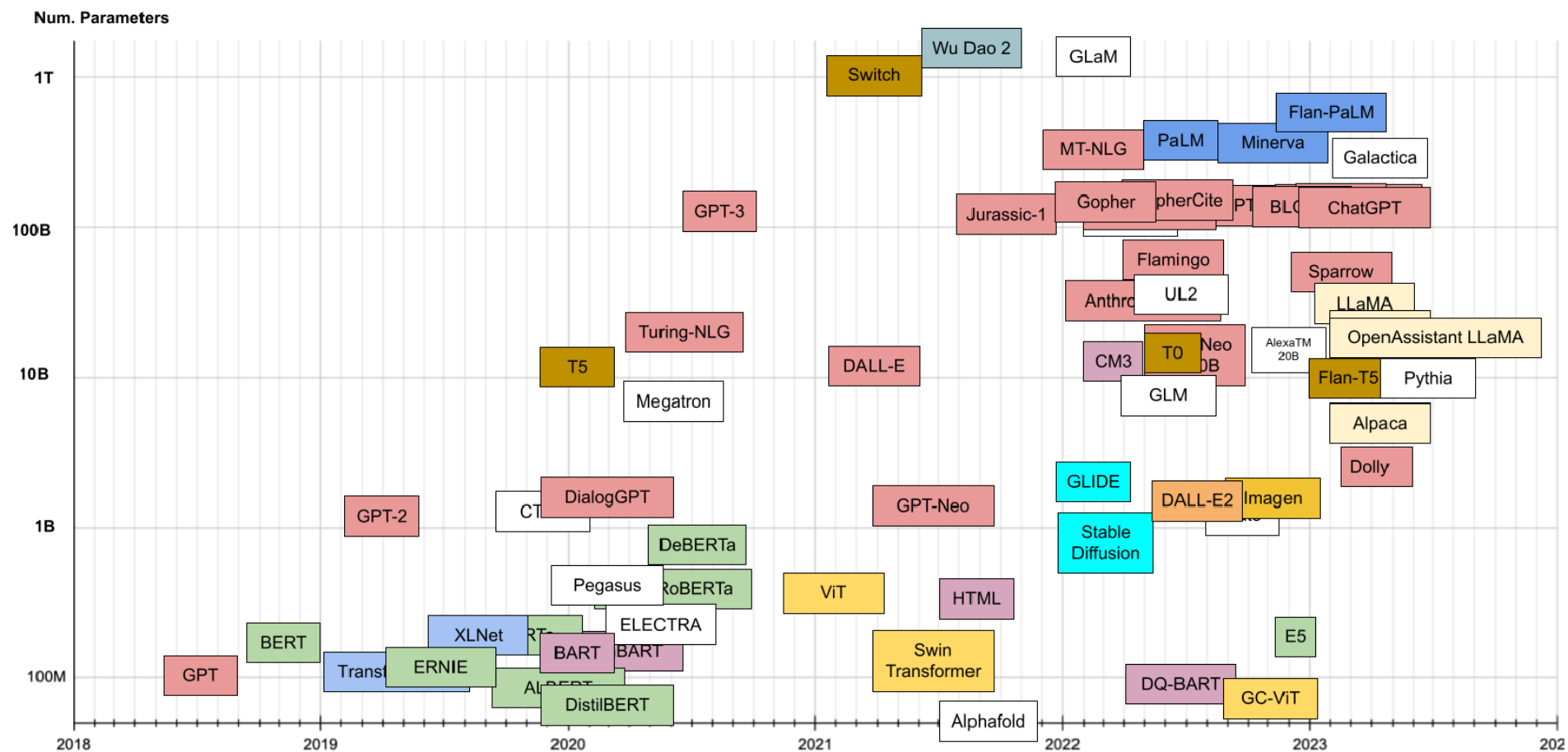
- Эволюция языковых моделей
- ELMo
- ERNIE
- ALBERT
- DistilBERT
- DeBERTa
- BART
- XLNet
- Longformer
- Switch Transformer
- FLAN
- RETRO

Эволюционное дерево

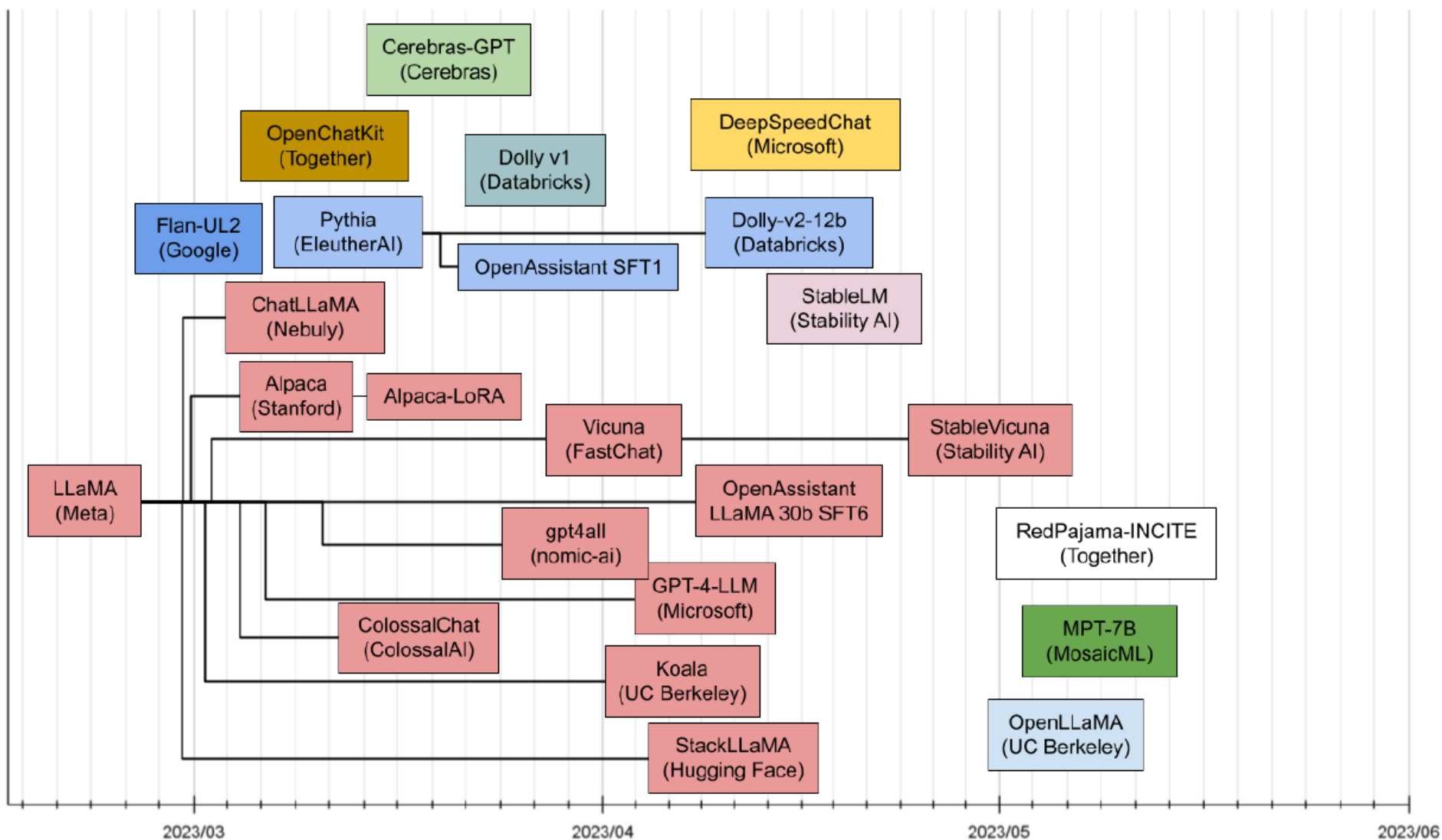
Yang J. et al. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond (2023)



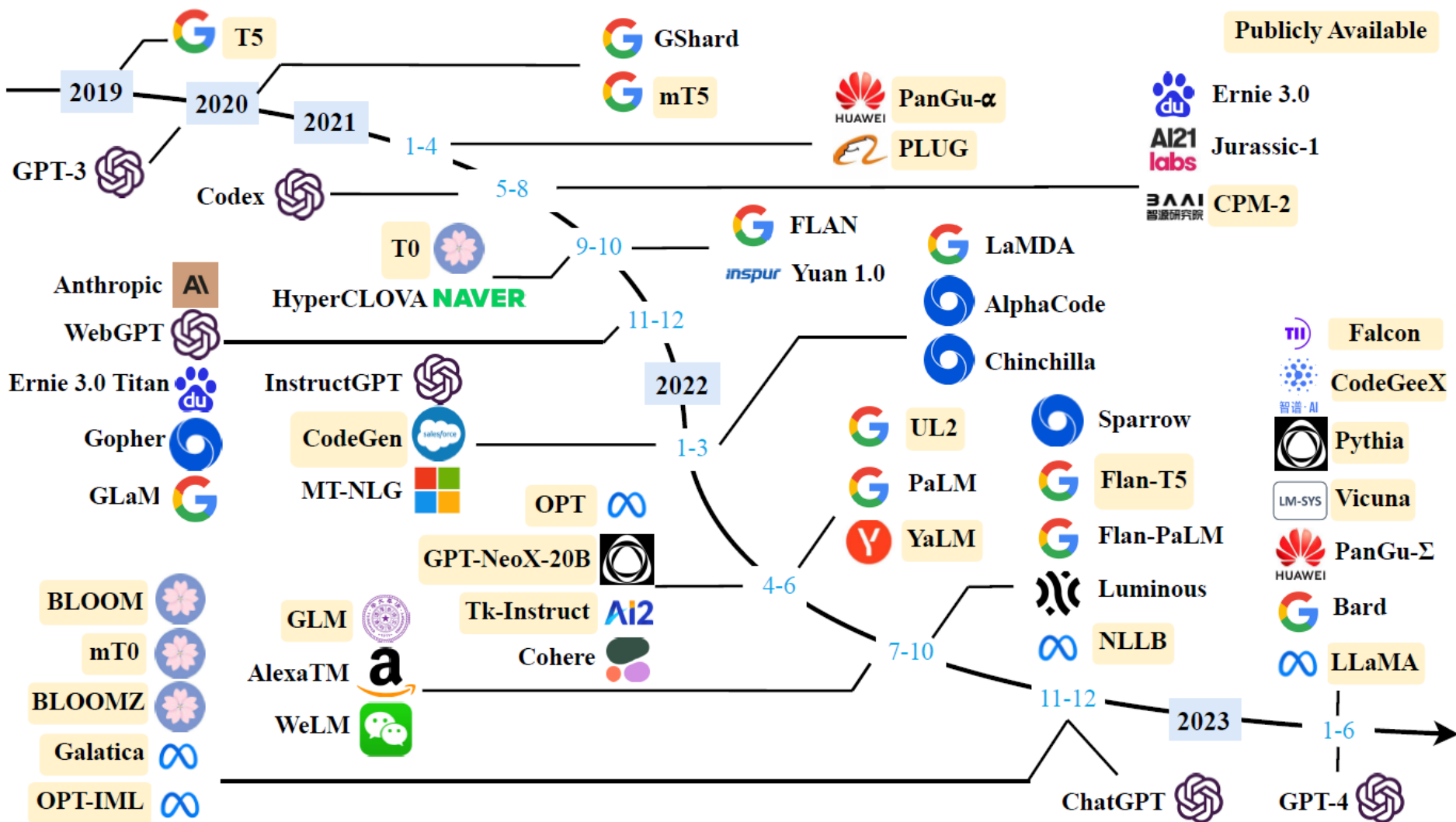
Временная шкала + параметры



Большие языковые модели



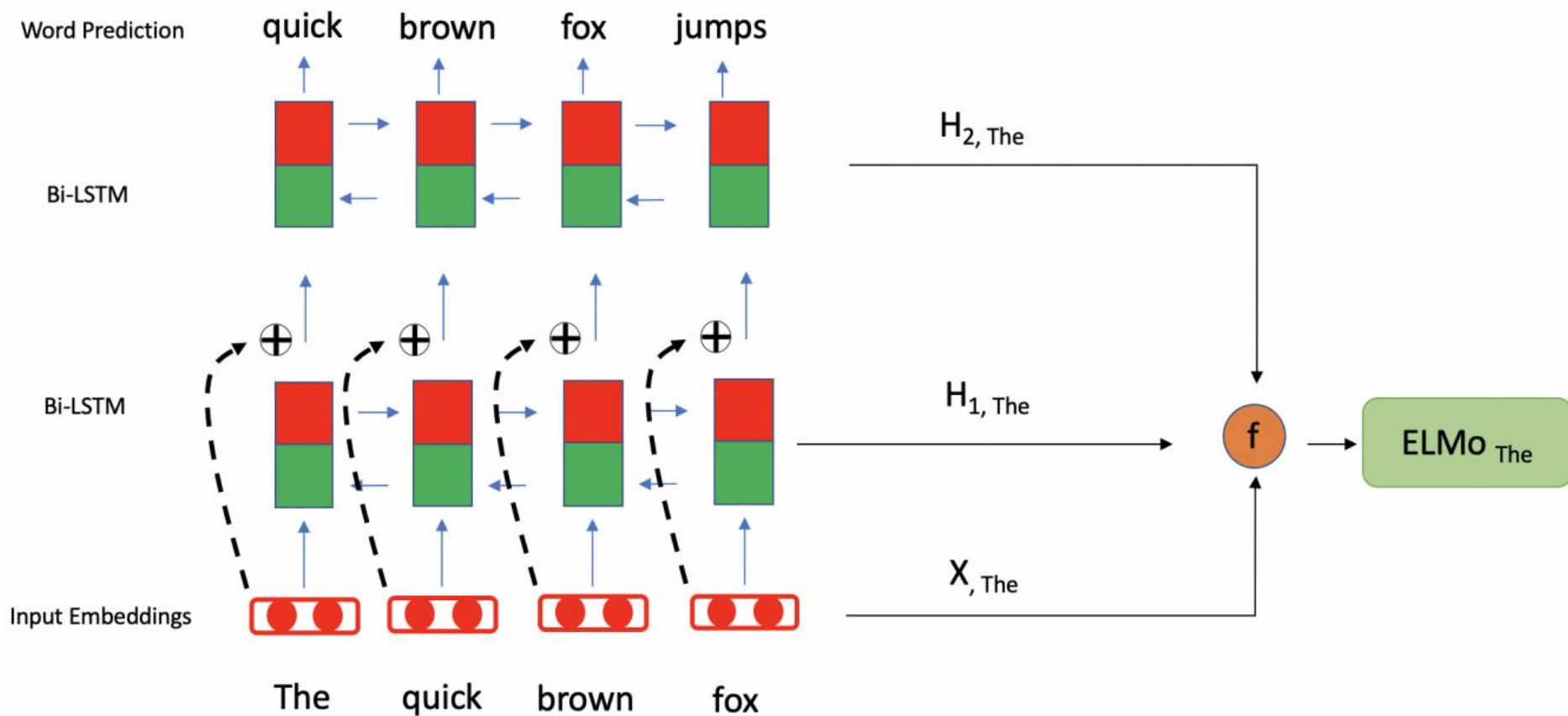
Большие языковые модели (>10B)



ELMo

- Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K., Zettlemoyer L. Deep Contextualized Word Representations // NAACL 2018
 - ELMo = Embeddings from Language Models
 - <https://aclanthology.org/N18-1202/>
 - <https://allenai.org/allennlp/software/elmo>
 - Allen Institute for Artificial Intelligence (февраль 2018)
- Основная идея: получение контекстных эмбеддингов слов на основе двунаправленной двуслойной LSTM и задачи языкового моделирования

ELMo: архитектура



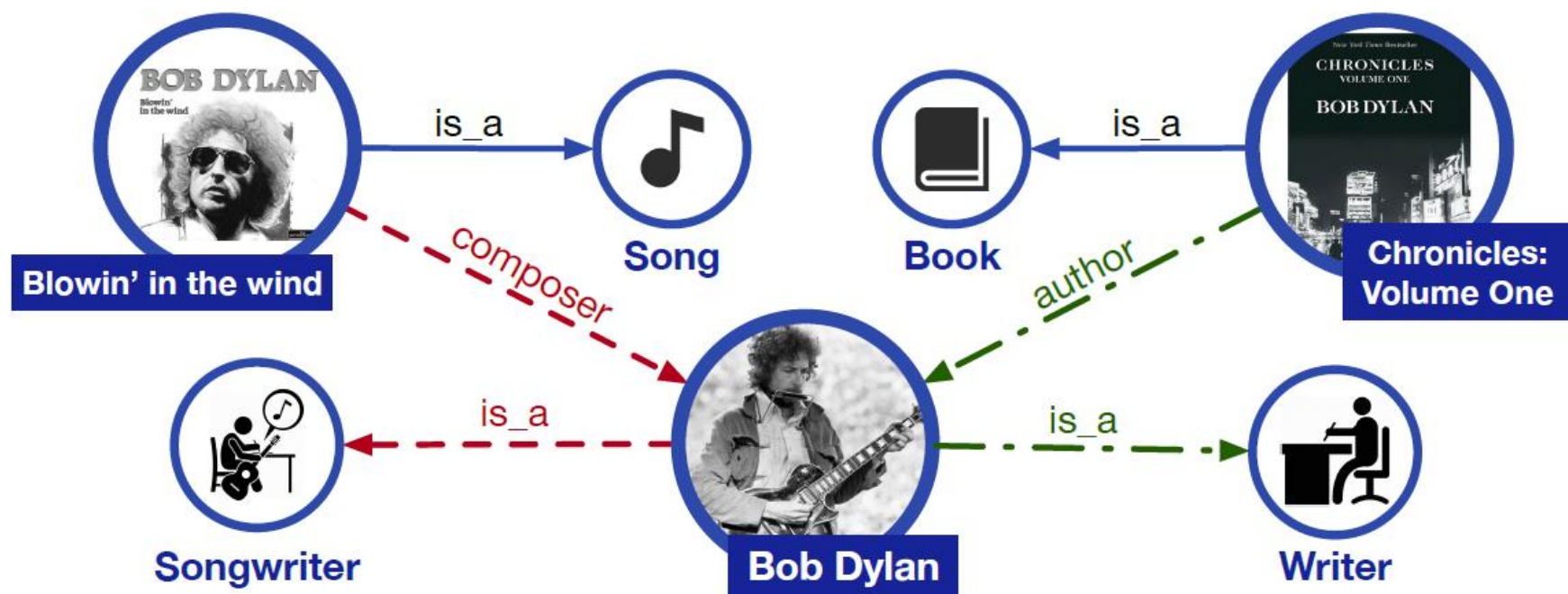
ELMo: результаты

TASK	PREVIOUS SOTA		OUR BASELINE	ELMo + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 $\pm$ 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 $\pm$ 0.19	90.15	92.22 $\pm$ 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 $\pm$ 0.5	3.3 / 6.8%

ERNIE

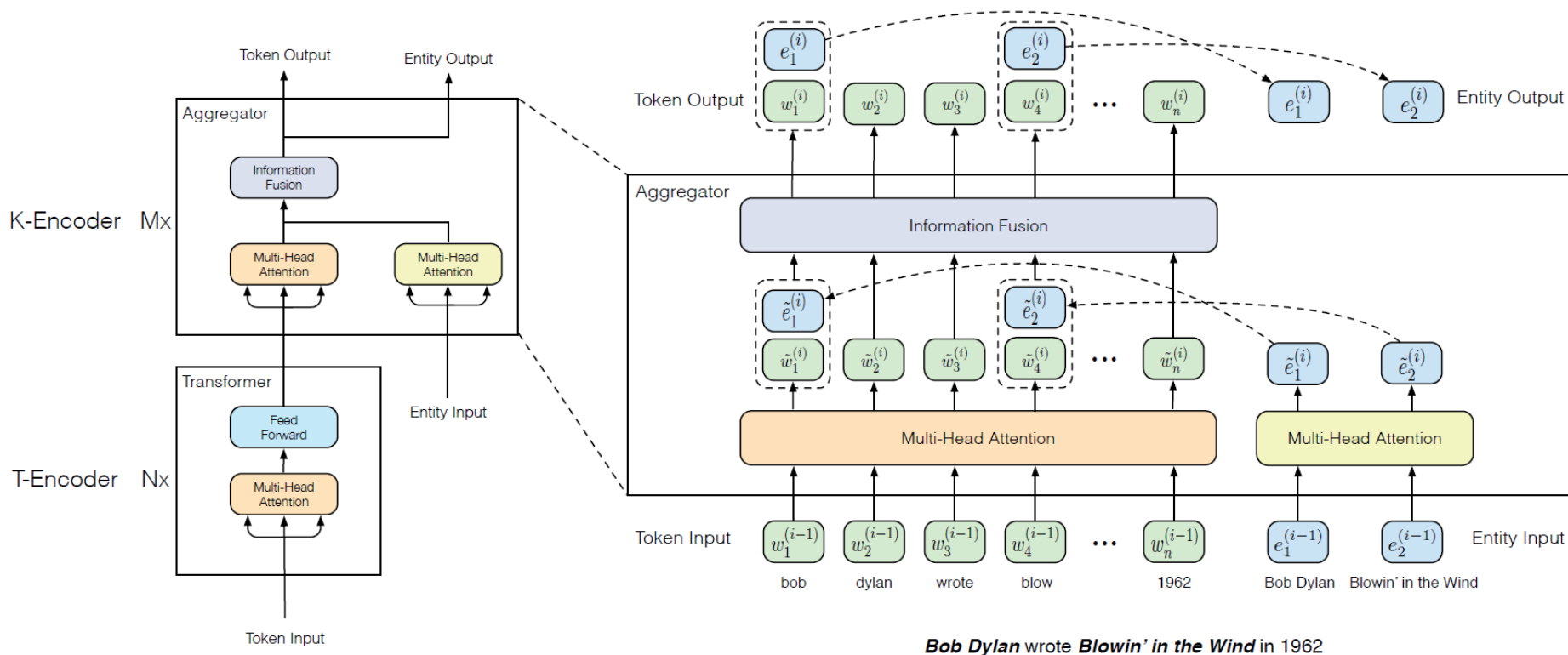
- Zhang Z., Han X., Liu Z., Jiang X., Sun M., Liu Q.
ERNIE: Enhanced Language Representation with Informative Entities // ACL 2019
 - ERNIE = **E**nhanced Language **R**epresentation**N** with Informative **E**ntities
 - <https://aclanthology.org/P19-1139/>
 - Tsinghua University (май 2019)
- Основная идея: включение в BERT графов знаний (knowledge graph)

ERNIE: идея



Bob Dylan wrote **Blowin' in the Wind** in 1962, and wrote **Chronicles: Volume One** in 2004.

ERNIE: архитектура



ERNIE: разметка типов сущностей и классификация отношений

FIGER

OpenEntity

Model	Acc.	Macro	Micro
NFGEC (Attentive)	54.53	74.76	71.58
NFGEC (LSTM)	55.60	75.15	71.73
BERT	52.04	75.16	71.63
ERNIE	57.19	76.51	73.39

Model	P	R	F1
NFGEC (LSTM)	68.80	53.30	60.10
UFET	77.40	60.60	68.00
BERT	76.37	70.96	73.56
ERNIE	78.42	72.90	75.56

Model	FewRel			TACRED		
	P	R	F1	P	R	F1
CNN	69.51	69.64	69.35	70.30	54.20	61.20
PA-LSTM	-	-	-	65.70	64.50	65.10
C-GCN	-	-	-	69.90	63.30	66.40
BERT	85.05	85.11	84.89	67.23	64.81	66.00
ERNIE	88.49	88.44	88.32	69.97	66.08	67.97

ERNIE: результаты на GLUE

Model	MNLI-(m/mm) 392k	QQP 363k	QNLI 104k	SST-2 67k
BERT _{BASE}	84.6/83.4	71.2	-	93.5
ERNIE	84.0/83.2	71.2	91.3	93.5

Model	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k
BERT _{BASE}	52.1	85.8	88.9	66.4
ERNIE	52.3	83.2	88.2	68.8

ALBERT

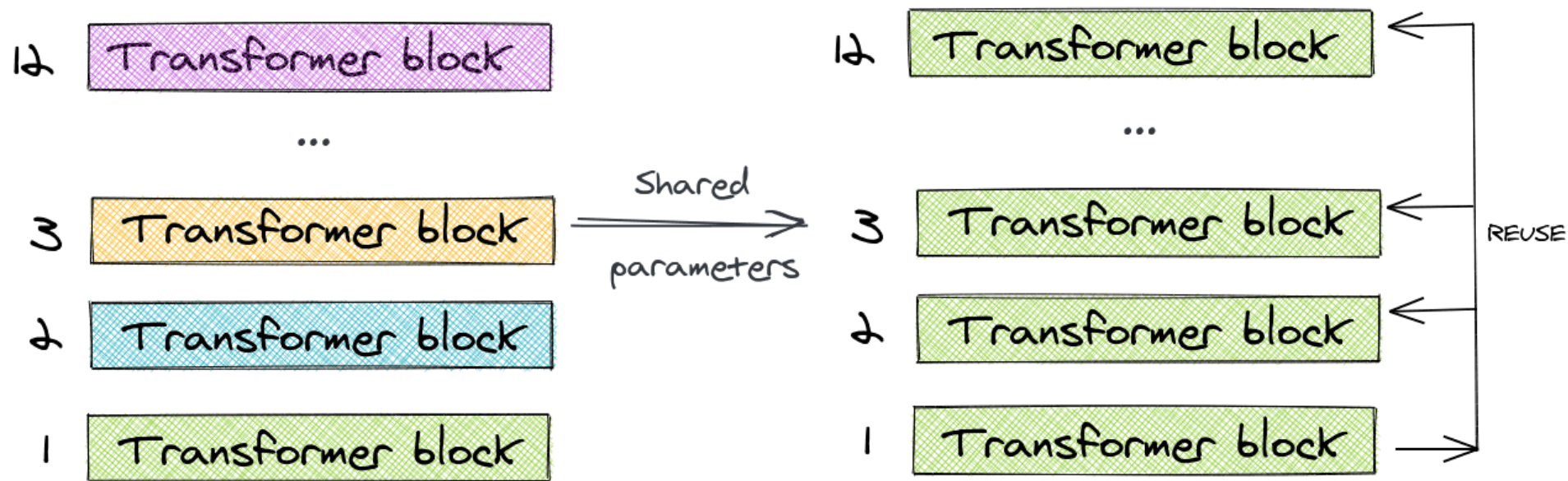
- Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations // ICLR 2020
 - https://iclr.cc/virtual_2020/poster_H1eA7AEtvS.html
 - <https://github.com/google-research/ALBERT>
 - Google (сентябрь 2019)
- Основная идея: легковесный BERT с таким же качеством
- [Amit Chaudhary. Visual Paper Summary: ALBERT](#)

ALBERT: Factorized embedding parameterization

- Размеры контекстно-независимых эмбеддингов (E) и контекстно-зависимых скрытых векторов (H) в BERT совпадают: $E = H$
- Однако эффективность BERT происходит из учета контекста, поэтому желательно $H \gg E$
- Также матрица эмбеддингов сильно зависит от размера словаря, поэтому желательно уменьшать E
- Декомпозиция матрицы эмбеддингов токенов на две небольшие матрицы

ALBERT: Cross-layer parameter sharing

- Разделение весов внимания и FFN между всеми слоями



ALBERT: Cross-layer parameter sharing

	Model	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base $E=768$	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base $E=128$	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

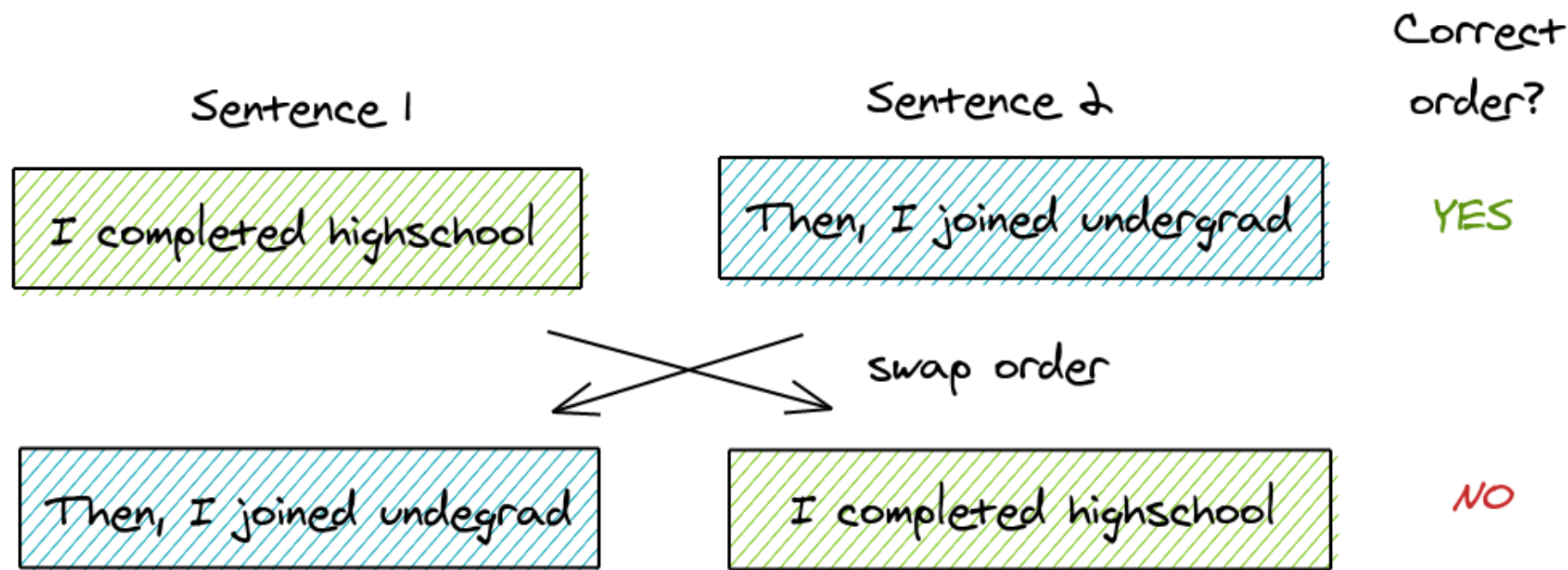
	Model	Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

ALBERT: Inter-sentence coherence loss

- В BERT задача NSP объединяет две задачи:
 - topic prediction – взяты ли два предложения из одного текста?
 - coherence prediction – следуют ли два предложения друг за другом?
- Задача topic prediction легче, поэтому NSP в целом не имеет смысла
- В ALBERT два предложения всегда берутся из одного документа (нет задачи topic prediction), но для негативного класса они переставляются – задача sentence-order prediction (SOP)

ALBERT: Inter-sentence coherence loss

ALBERT:



ALBERT: Inter-sentence coherence loss

SP tasks	Intrinsic Tasks			Downstream Tasks					Avg
	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	91.1	62.3	79.2
SOP	54.0	78.9	86.5	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1

ALBERT: GLUE и сравнение с BERT

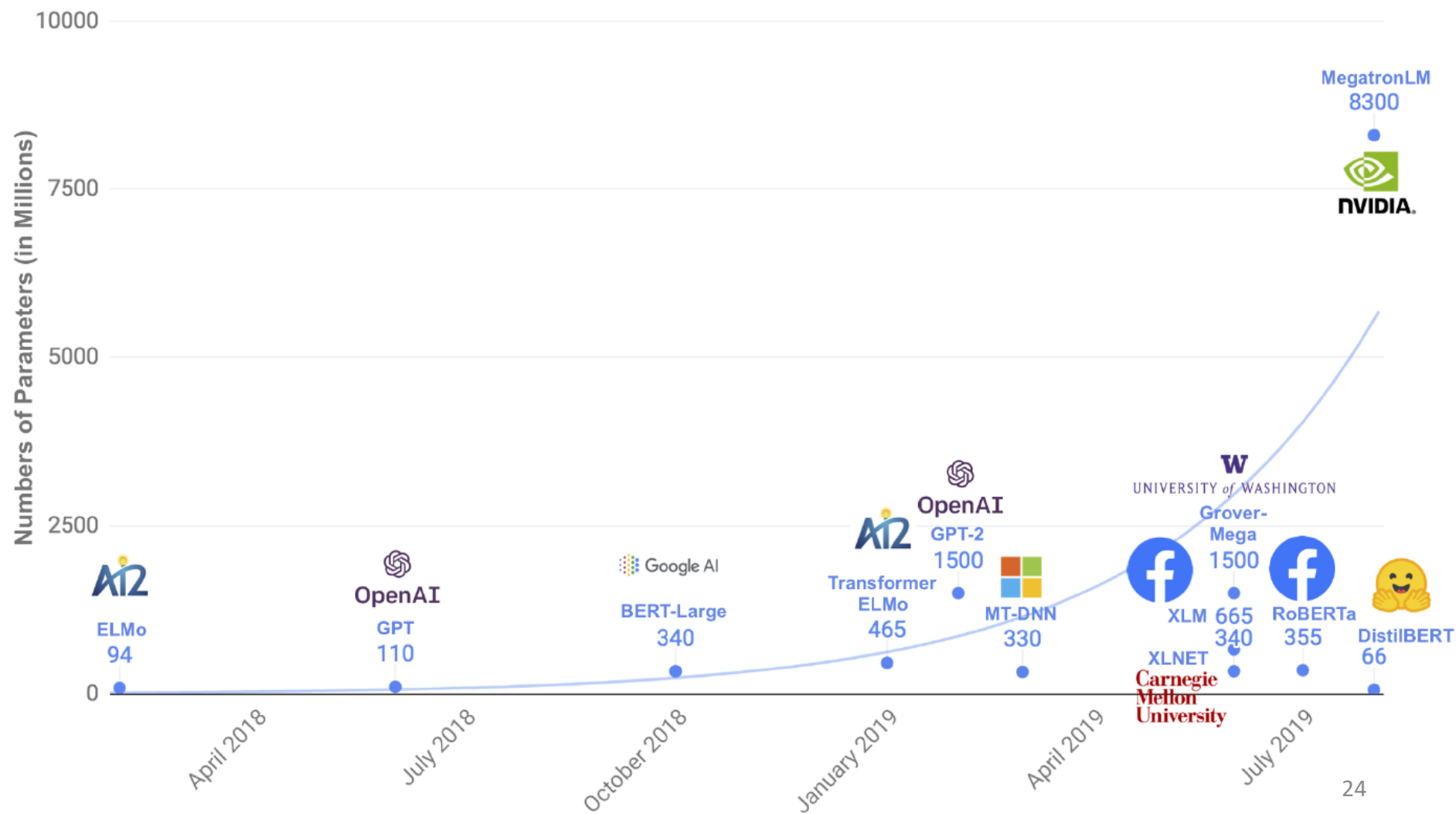
Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS	WNLI	Avg
<i>Single-task single models on dev</i>										
BERT-large	86.6	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-	-
XLNet-large	89.8	93.9	91.8	83.8	95.6	89.2	63.6	91.8	-	-
RoBERTa-large	90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	-	-
ALBERT (1M)	90.4	95.2	92.0	88.1	96.8	90.2	68.7	92.7	-	-
ALBERT (1.5M)	90.8	95.3	92.2	89.2	96.9	90.9	71.4	93.0	-	-
<i>Ensembles on test (from leaderboard as of Sept. 16, 2019)</i>										
ALICE	88.2	95.7	90.7	83.5	95.2	92.6	69.2	91.1	80.8	87.0
MT-DNN	87.9	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0	87.6
XLNet	90.2	98.6	90.3	86.3	96.8	93.0	67.8	91.6	90.4	88.4
RoBERTa	90.8	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0	88.5
Adv-RoBERTa	91.1	98.8	90.3	88.7	96.8	93.1	68.0	92.4	89.0	88.8
ALBERT	91.3	99.2	90.5	89.2	97.1	93.4	69.1	92.5	91.8	89.4

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	4.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	5.6x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	1.7x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	0.6x
	xxlarge	235M	94.1/88.3	88.1/85.1	88.0	95.2	82.3	88.7	0.3x

DistilBERT

- Sanh V., Debut L., Chaumond J., Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter // arXiv:1910.01108
 - <https://arxiv.org/abs/1910.01108>
 - Hugging Face (октябрь 2019)
- Основная идея: knowledge distillation для BERT
 - Geoffrey Hinton, Oriol Vinyals, Jeff Dean. Distilling the Knowledge in a Neural Network // [arXiv:1503.02531](https://arxiv.org/abs/1503.02531)
 - Обучение небольшой модели («студента») на ответах большой модели («учителя»)

DistilBERT



DistilBERT: функция ошибки

- Учет всего распределения «учителя»:

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

где t_i – вероятности ответов «учителя»,

s_i – вероятности ответов «студента»

- Моделирование всего распределения позволяет лучше обобщаться
 - I think this is the beginning of a beautiful [MASK]
 - [MASK] – **day, life**, future, story, world, ...

DistilBERT: функция ошибки

- Тройная функция ошибки (линейная комбинация):
 - Кросс-энтропия для ответов «учителя» L_{ce}
 - Кросс-энтропия для MLM
 - cosine embedding loss – разница между скрытыми векторами «учителя» и «студента»

DistilBERT

- Архитектура:
 - убрали token-type embeddings и pooler
 - сократили количество слоев в два раза
 - размерность векторов сохранили
- Инициализация:
 - слоями «учителя» (один слой из двух)
- Обучение:
 - Датасет как в BERT: English Wikipedia и Book Corpus
 - 8 V100 16 GB – 90 часов
 - RoBERTa: 1024 V100 32 GB – 24 часа

DistilBERT: результаты

- На 40% меньше, на 60% быстрее, 97% качества

Model	Score	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

DeBERTa

- He P., Liu X., Gao J., Chen W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention
// ICLR 2021
 - Microsoft (июнь 2020)
 - <https://github.com/microsoft/DeBERTa>
 - Третья версия – ICLR 2023
 - <https://openreview.net/forum?id=sE7-XhLxHA>
- Основная идея: лучше учитывать позиции токенов
 - <https://www.youtube.com/watch?v=c6A33Fg5Ns>

DeBERTa: Disentangled attention

- В BERT каждый токен представляется суммой эмбединга токена и эмбединга позиции этого токена
- В DeBERTa каждый токен представляется двумя отдельными векторами - эмбедингом токена и эмбедингом *относительной* позиции этого токена
- Пример: слова «deer» и «learning» связаны по-разному, если они встречаются вместе или достаточно далеко друг от друга

$$A_{i,j} = \{H_i, P_{i|j}\} \times \{H_j, P_{j|i}\}^T =$$

$$= H_i H_j^T + H_i P_{j|i}^T + P_{i|j} H_j^T + P_{i|j} P_{j|i}^T$$

content-to-content

content-to-position

position-to-content

position-to-position

DeBERTa: Disentangled attention

- В BERT каждый токен представляется суммой эмбединга токена и эмбединга позиции этого токена
- В DeBERTa каждый токен представляется двумя отдельными векторами - эмбедингом токена и эмбедингом *относительной* позиции этого токена
- Пример: слова «deer» и «learning» связаны по-разному, если они встречаются вместе или достаточно далеко друг от друга

$$A_{i,j} = \{H_i, P_{i|j}\} \times \{H_j, P_{j|i}\}^T =$$

$$= H_i H_j^T + H_i P_{j|i}^T + P_{i|j} H_j^T + P_{i|j} P_{j|i}^T$$

content-to-content
(было ранее)

content-to-position
(было ранее)

position-to-content
(DeBERTa)

position-to-position
(не используется)

DeBERTa: Enhanced mask decoder

- Кроме относительных позиций важны и абсолютные:

A new store opened beside the new mall

- Для решения задачи Masked Language Modeling (MLM) абсолютная позиция токенов учитывается после выходного слоя Transformer перед Softmax
- В BERT эмбединг абсолютной позиции токена складывался с эмбедингом самого токена на входе Transformer

DeBERTa: результаты на GLUE

Model	CoLA Mcc	QQP Acc	MNLI-m/mm Acc	SST-2 Acc	STS-B Corr	QNLI Acc	RTE Acc	MRPC Acc	Avg.
BERT _{large}	60.6	91.3	86.6/-	93.2	90.0	92.3	70.4	88.0	84.05
RoBERTa _{large}	68.0	92.2	90.2/90.2	96.4	92.4	93.9	86.6	90.9	88.82
XLNet _{large}	69.0	92.3	90.8/90.8	97.0	92.5	94.9	85.9	90.8	89.15
ELECTRA _{large}	69.1	92.4	90.9/-	96.9	92.6	95.0	88.0	90.8	89.46
DeBERTa _{large}	70.5	92.3	91.1/91.1	96.8	92.8	95.3	88.3	91.9	90.00

GLUE

Rank Name		Model	URL	Score
1	Microsoft Alexander v-team	Turing ULR v6		91.3
2	JDExplore d-team	Vega v1		91.3
3	Microsoft Alexander v-team	Turing NLR v5		91.2
4	DIRL Team	DeBERTa + CLEVER		91.1
5	ERNIE Team - Baidu	ERNIE		91.1
6	AliceMind & DIRL	StructBERT + CLEVER		91.0
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8
8	HFL iFLYTEK	MacALBERT + DKM		90.7
9	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6
10	T5 Team - Google	T5		90.3

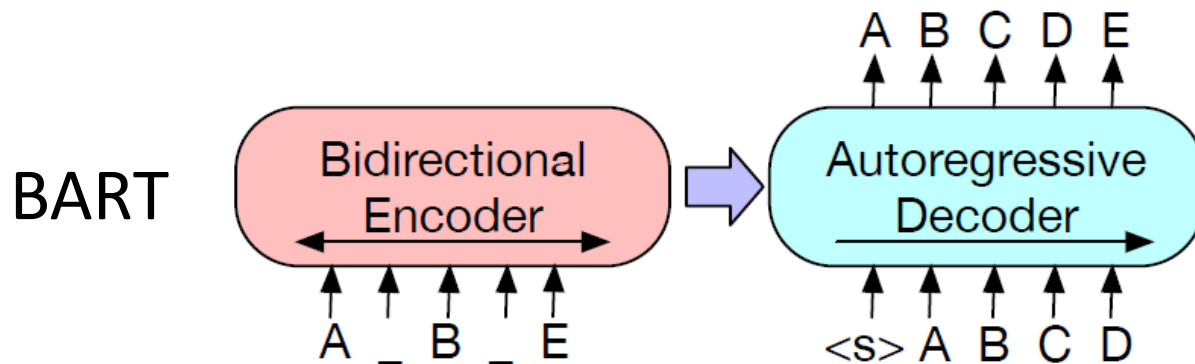
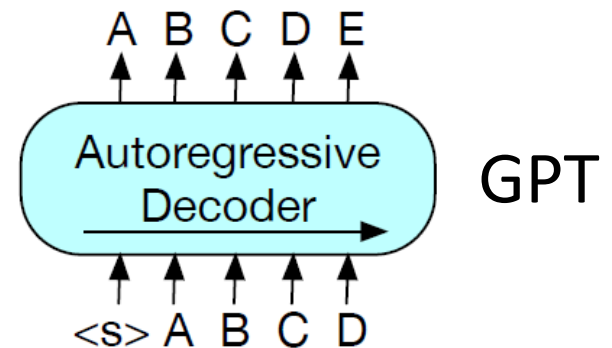
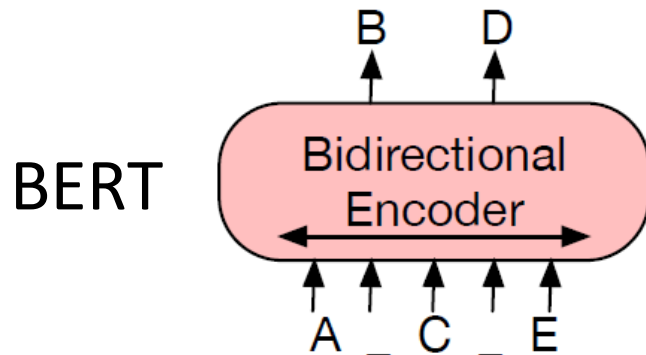
SuperGLUE

Rank	Name	Model	Score
1	JDExplore d-team	Vega v2	91.3
2	Liam Fedus	ST-MoE-32B	91.2
3	Microsoft Alexander v-team	Turing NLR v5	90.9
4	ERNIE Team - Baidu	ERNIE 3.0	90.6
5	Yi Tay	PaLM 540B	90.4
6	Zirui Wang	T5 + UDG, Single Model (Google Brain)	90.4
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	90.3
8	SuperGLUE Human Baselines	SuperGLUE Human Baselines	89.8
9	T5 Team - Google	T5	89.3
10	SPoT Team - Google	Frozen T5 1.1 + SPoT	89.2

BART

- Lewis M., Liu Y., Goyal N., Ghazvininejad M., Mohamed A., Levy O., Stoyanov V., Zettlemoyer L.
BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension // ACL 2020
 - BART – **B**idirectional and **A**uto-**R**egressive **T**ransformers
 - <https://aclanthology.org/2020.acl-main.703/>
 - Facebook AI (октябрь 2019)
- Основная идея: объединение преимуществ BERT и GPT – классификация и генерация на одной модели

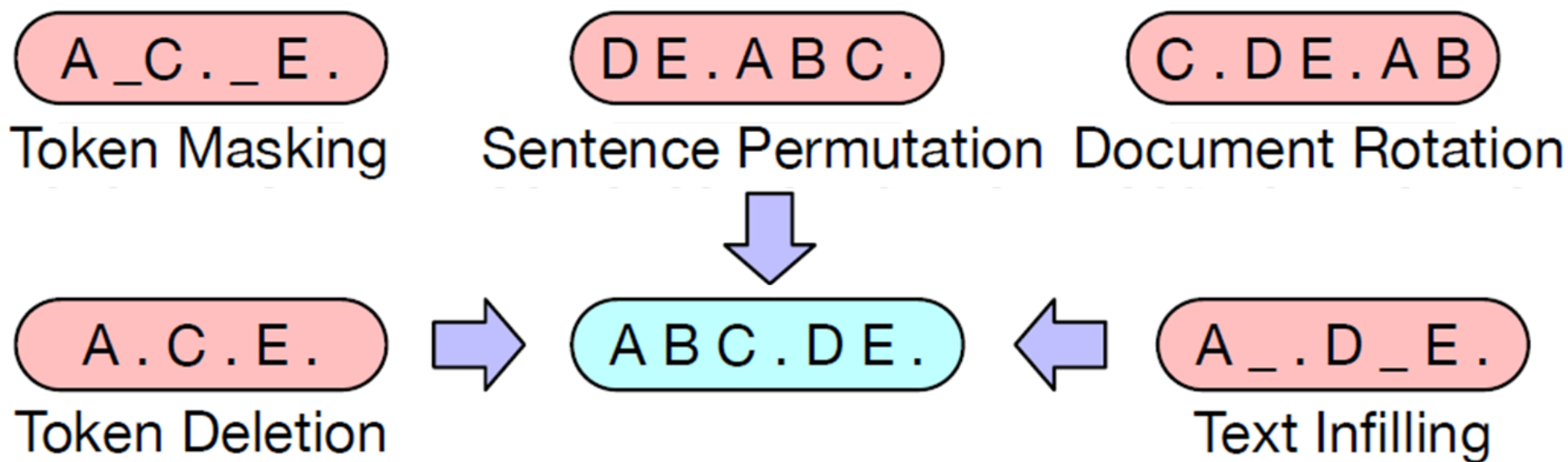
BART: архитектура



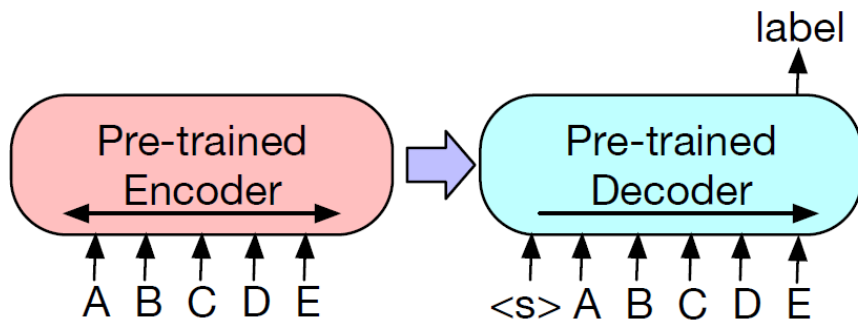
- Base: 6 слоев
- Large: 12 слоев

BART: Pretraining

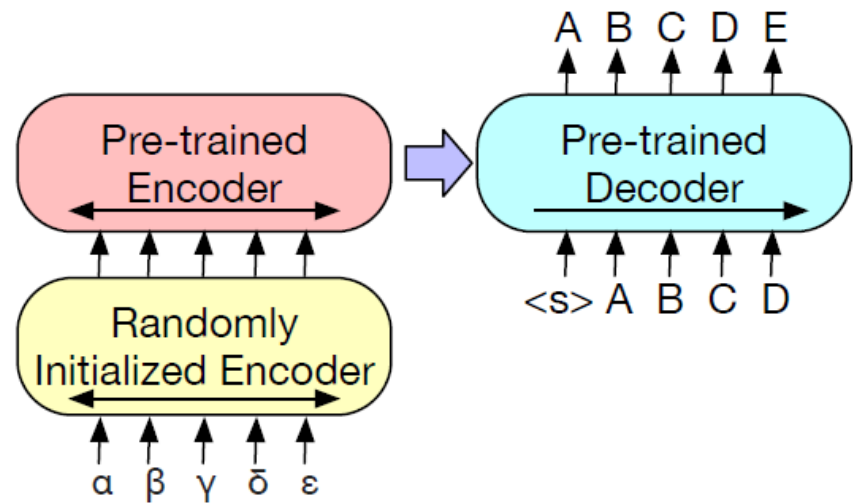
- Допускается любое искажение входного текста
- В пределе (когда нет информации о входе) – традиционная языковая модель



BART: Fine-tuning



Классификация



Машинный перевод

BART: результаты на GLUE

	MNLI	SST	QQP	QNLI	STS-B	RTE	MRPC	CoLA
	m/mm	Acc	Acc	Acc	Acc	Acc	Acc	Mcc
BERT	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	90.2/90.2	96.4	92.2	94.7	92.4	86.6	90.9	68.0
BART	89.9/90.1	96.6	92.5	94.9	91.2	87.0	90.4	62.8

BART: результаты для реферирования

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
ROBERTASHARE (Rothe et al., 2019)	40.31	18.91	37.62	41.45	18.79	33.90
BART	44.16	21.28	40.90	45.14	22.27	37.25

XLNet

- Yang Z., Dai Z., Yang Y., Carbonell J., Salakhutdinov R., Le Q.V.
XLNet: Generalized Autoregressive Pretraining for Language Understanding // NeurIPS 2019
 - Название от Transformer-XL
 - [NeurIPS Proceedings](#)
 - Carnegie Mellon University, Google (июнь 2019)
- Основная идея: включение двунаправленности в декодере за счет перестановок при вычислении вероятности токена
 - Также включили рекуррентность по сегментам и относительное позиционное кодирование из Transformer-XL

XLNet: результаты

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
<i>Single-task single models on dev</i>									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	-
XLNet	90.8/90.8	94.9	92.3	85.9	97.0	90.8	69.0	92.5	-
<i>Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)</i>									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	90.9/90.9[†]	99.0[†]	90.4[†]	88.5	97.1[†]	92.9	70.2	93.0	92.5

Longformer

- Beltagy I., Peters M.E., Cohan A. Longformer: The Long-Document Transformer // arXiv:2004.05150
 - <https://github.com/allenai/longformer>
 - Allen Institute for Artificial Intelligence (апрель 2020)
- Основная идея: обработка длинных последовательностей за счет внимания линейной сложности
 - Реализуется на основе специальных паттернов
- Может применяться для энкодера, декодера и энкодера-декодера

Longformer: результаты

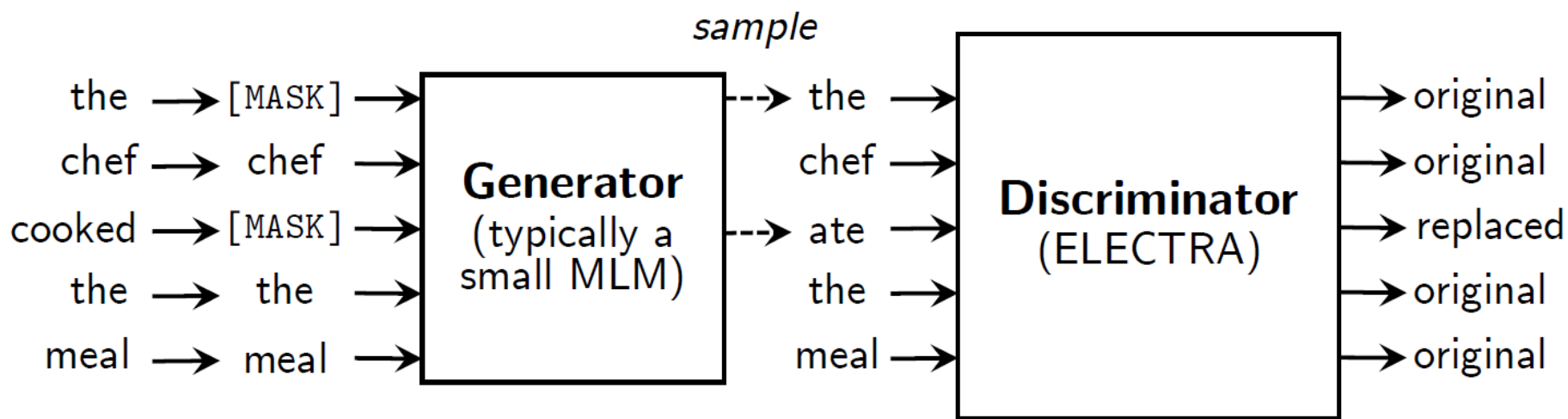
Model	QA			Coref.	Classification	
	WikiHop	TriviaQA	HotpotQA	OntoNotes	IMDB	Hyperpartisan
RoBERTa-base	72.4	74.3	63.5	78.4	95.3	87.4
Longformer-base	75.0	75.2	64.4	78.6	95.7	94.8

ELECTRA

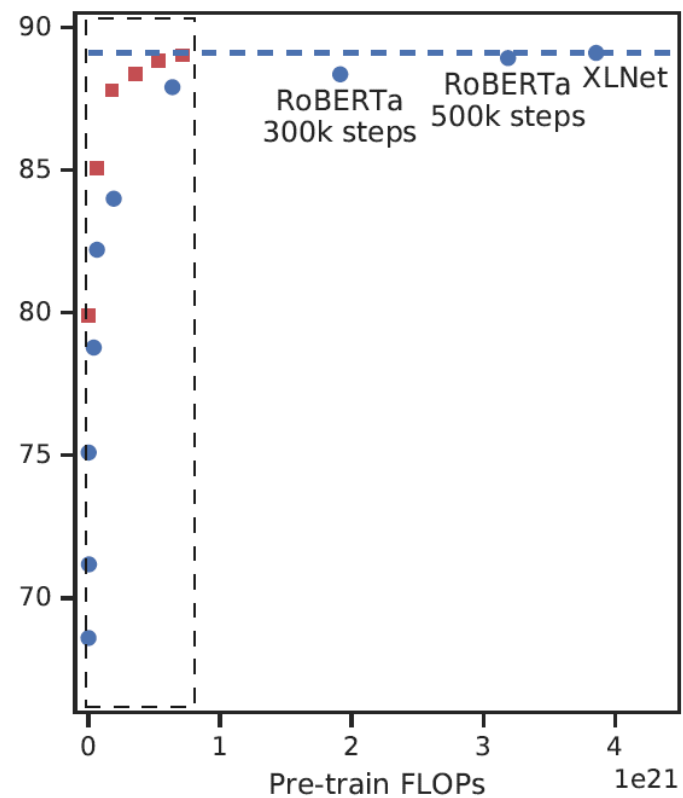
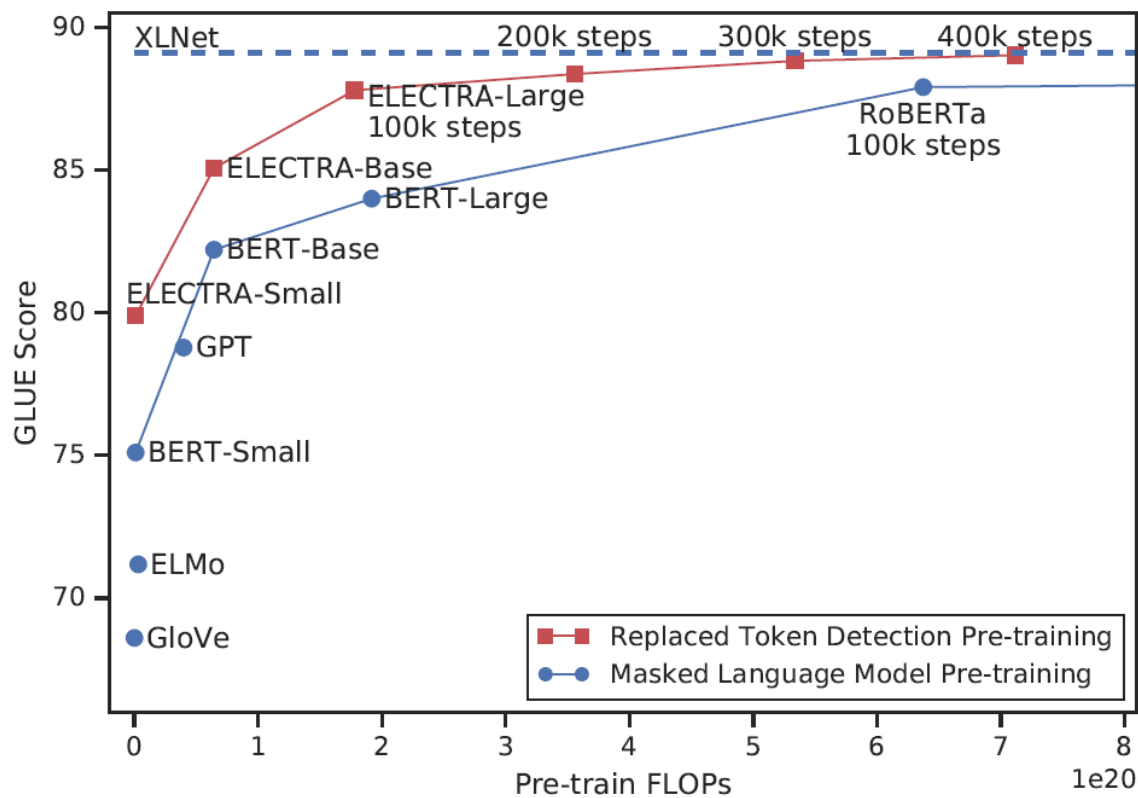
- Clark K., Luong M.-T., Le Q.V., Manning C.D. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators // ICLR 2020
 - ELECTRA = **E**fficiently **L**earning an **E**ncoder that **C**lassifies **T**oken **R**emplacements **A**ccurately
 - <https://openreview.net/pdf?id=r1xMH1BtvB>
 - Stanford University, Google (март 2020)
- Основная идея: вместо MLM решается задача Replaced Token Detection – специальная модель генерирует заменяющие токены, а обучаемая модель пытается определить, какой токен заменен

ELECTRA

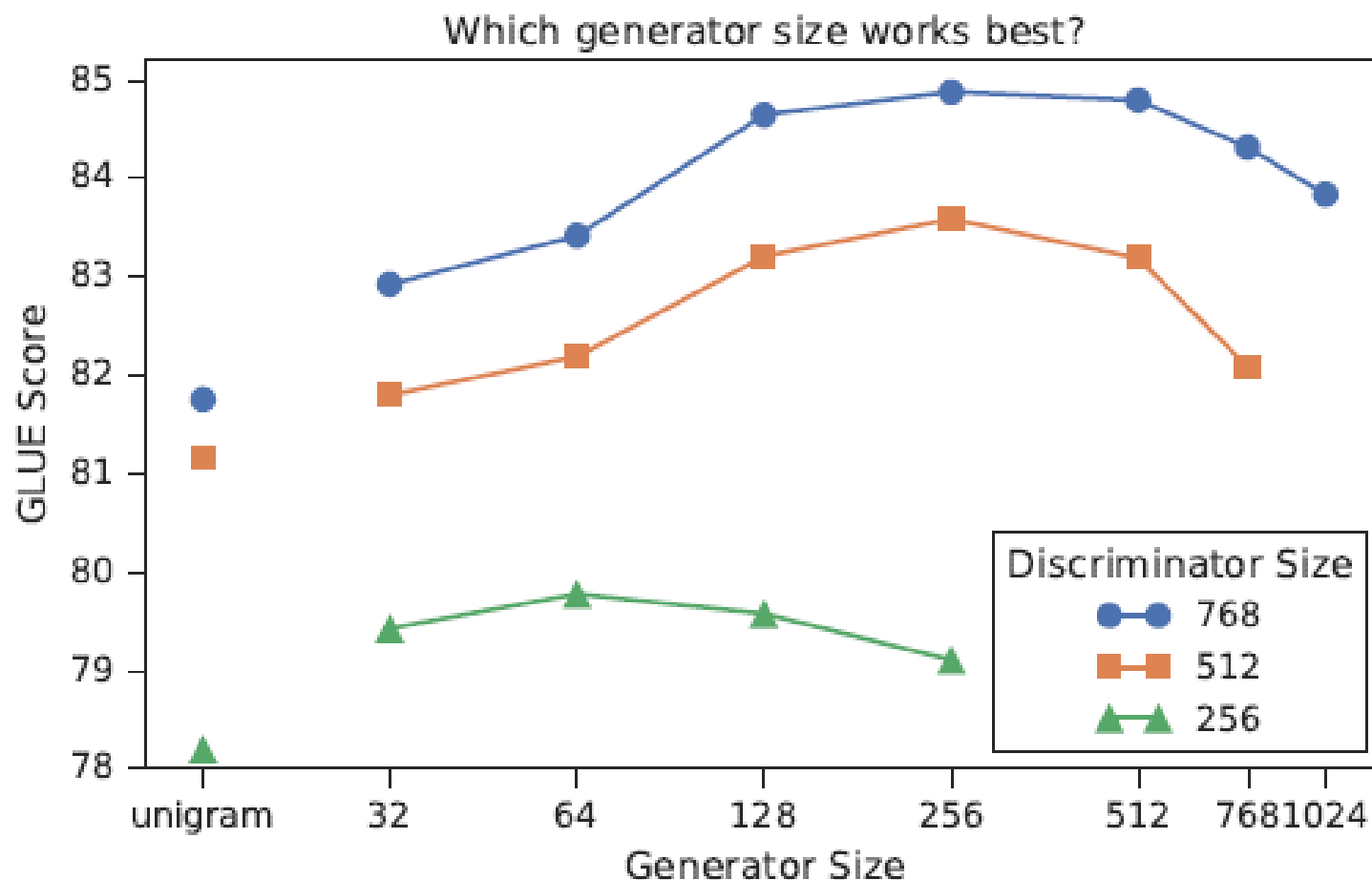
- Подход:



ELECTRA: результаты



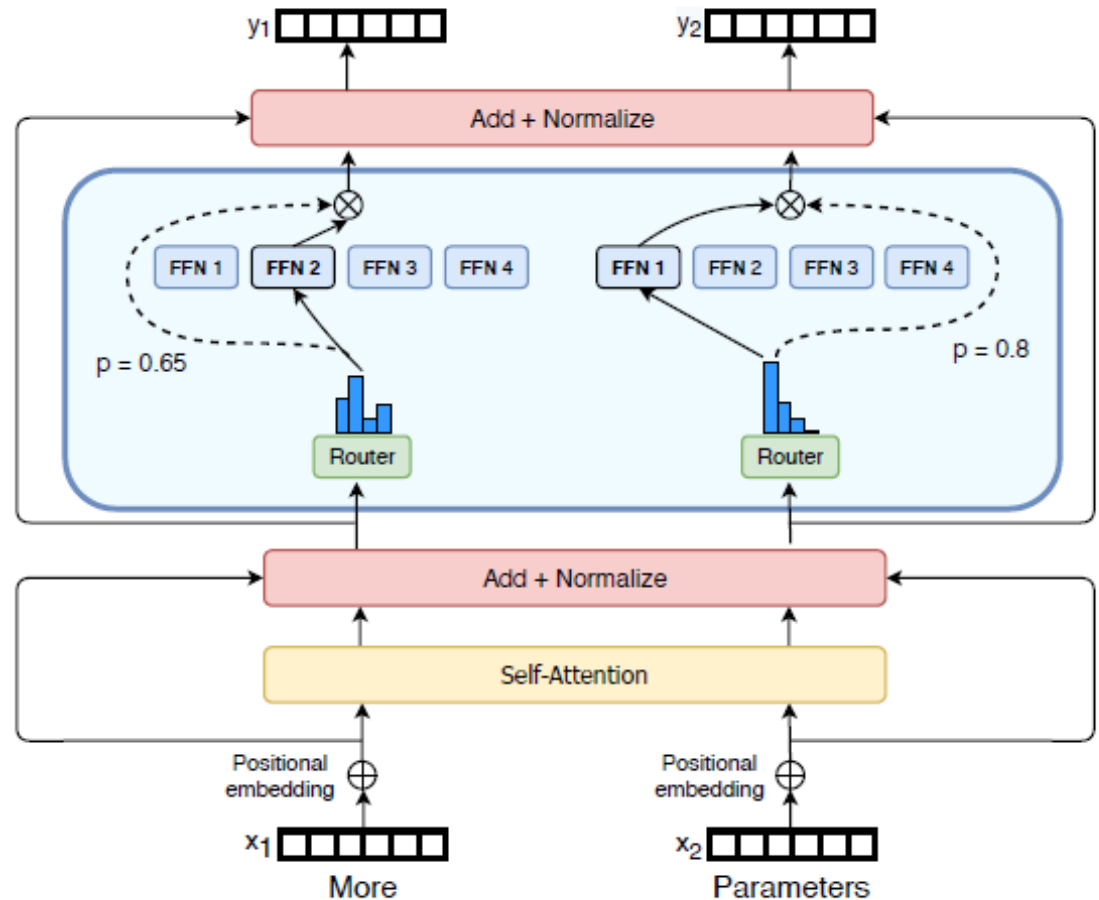
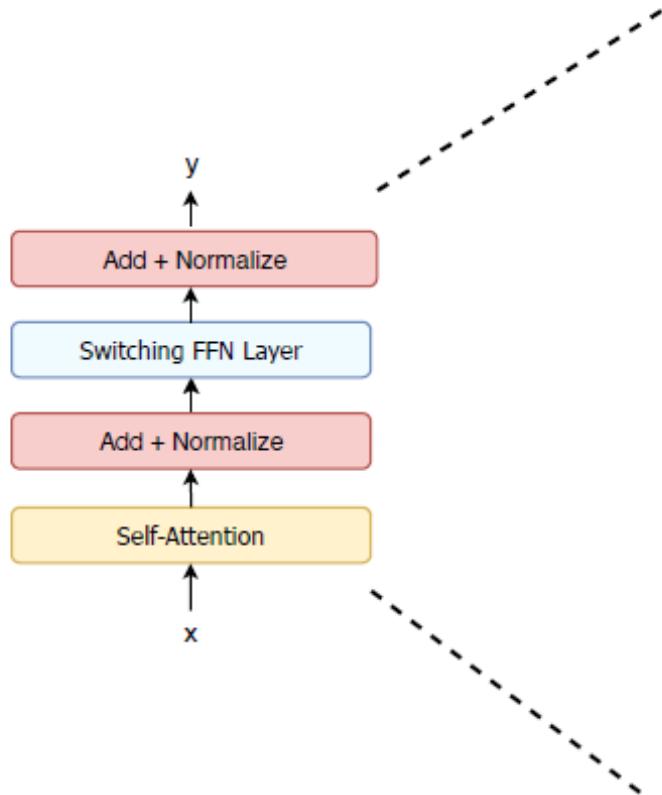
ELECTRA: результаты



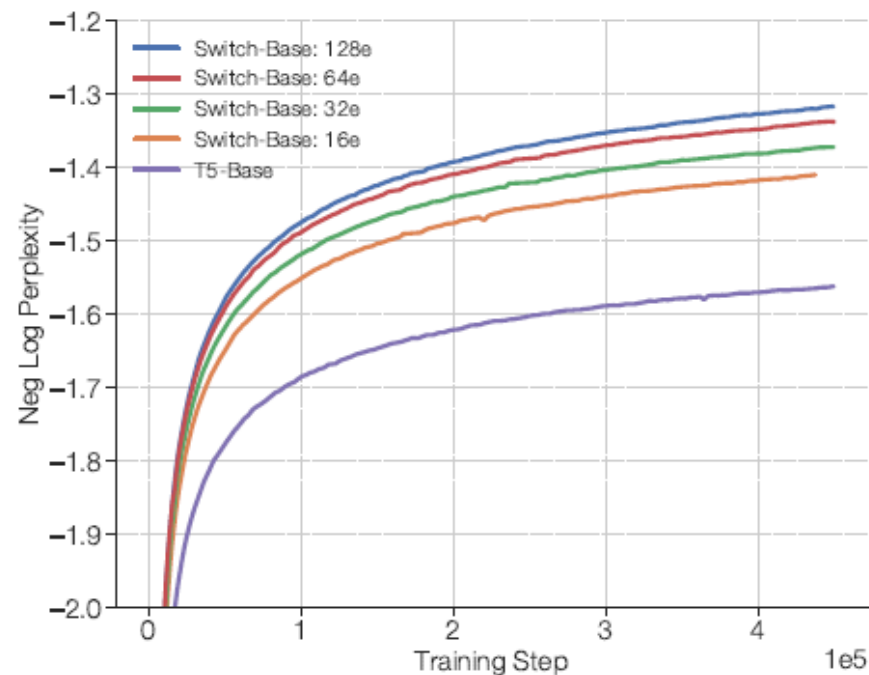
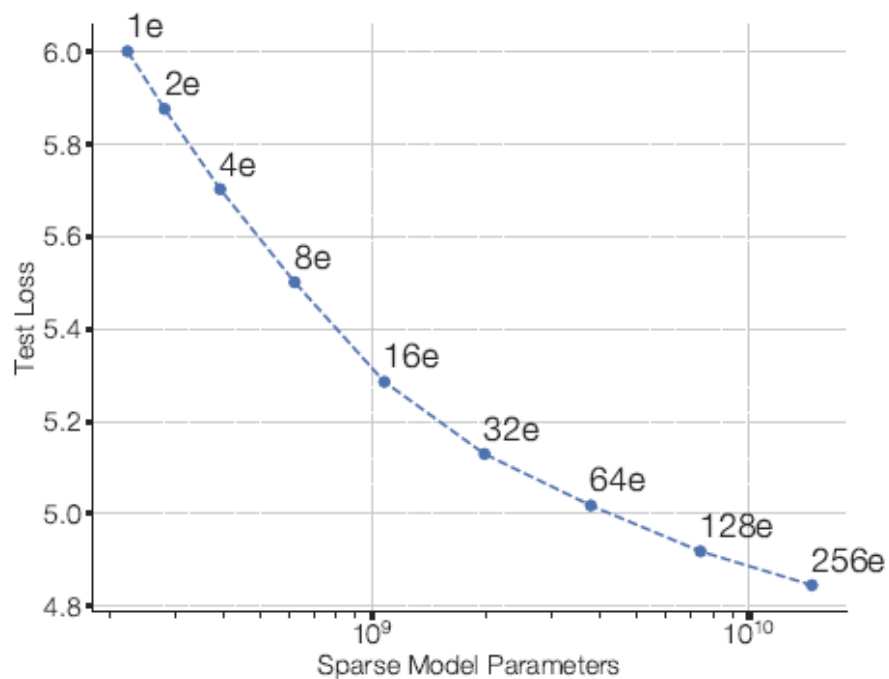
Switch Transformer

- Fedus W., Zoph B., Shazeer N. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity // Journal of Machine Learning Research, 2022
 - <https://jmlr.org/papers/v23/21-0998.html>
 - Google (январь 2021)
- Основная идея: разреженная активация – для каждого примера работает подмножество весов сети (идея Mixture of Experts)
 - 1 триллион параметров, Encoder-Decoder (T5)

Switch Transformer: архитектура



Switch Transformer



Switch Transformer: результаты

Model	GLUE	SQuAD	SuperGLUE	Winogrande (XL)
T5-Base	84.3	85.5	75.1	66.6
Switch-Base	86.7	87.2	79.5	73.3
T5-Large	87.8	88.1	82.7	79.1
Switch-Large	88.5	88.6	84.7	83.0

Model	XSum	ANLI (R3)	ARC Easy	ARC Chal.
T5-Base	18.7	51.8	56.7	35.5
Switch-Base	20.3	54.0	61.3	32.8
T5-Large	20.9	56.6	68.8	35.5
Switch-Large	22.3	58.6	66.0	35.5

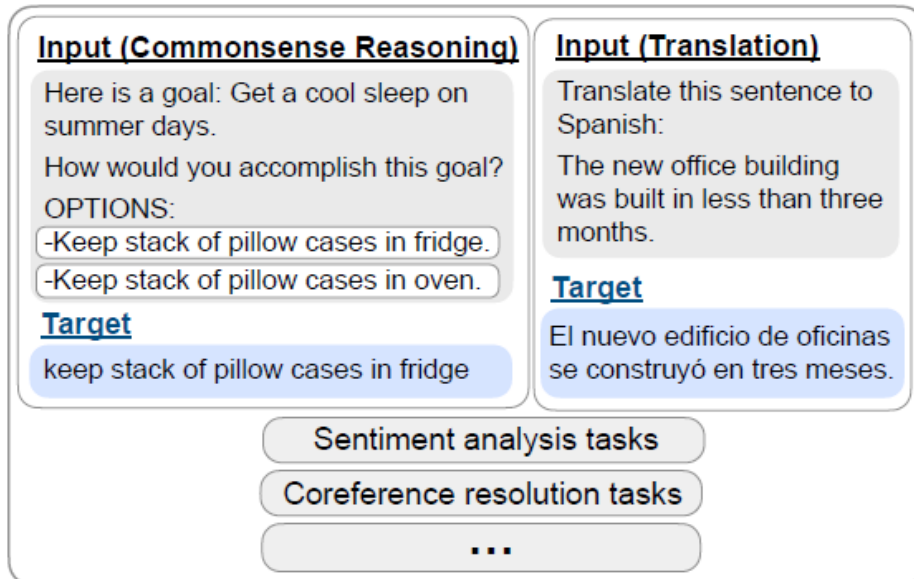
Model	CB Web QA	CB Natural QA	CB Trivia QA
T5-Base	26.6	25.8	24.5
Switch-Base	27.4	26.8	30.7
T5-Large	27.7	27.6	29.5
Switch-Large	31.3	29.5	36.9

FLAN

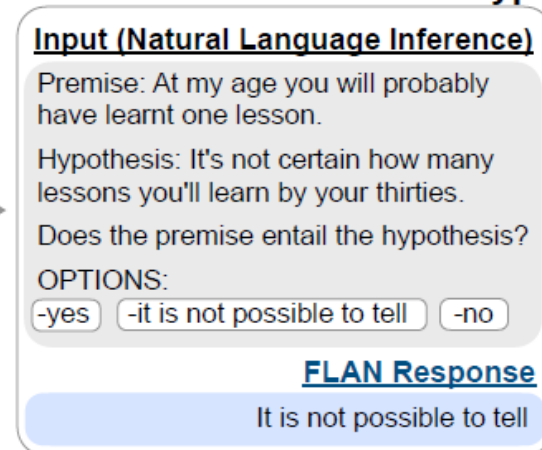
- Wei J., Bosma M., Zhao V.Y., Guu K., Yu A.W., Lester B., Du N., Dai A.M., Le Q.V. Finetuned Language Models Are Zero-Shot Learners // ICLR 2022
 - FLAN = Finetuned Language Net
 - Google (сентябрь 2021)
- Основная идея: инструкции по множеству задач (60 датасетов) для дообучения большой модели
 - GPT, 137 миллиардов параметров

FLAN: подход

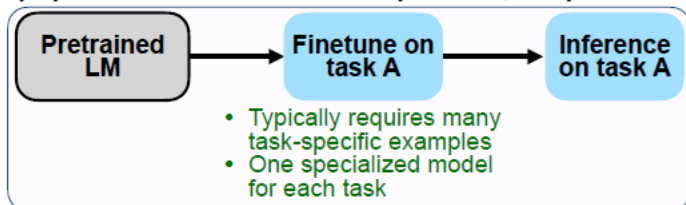
Finetune on many tasks (“instruction-tuning”)



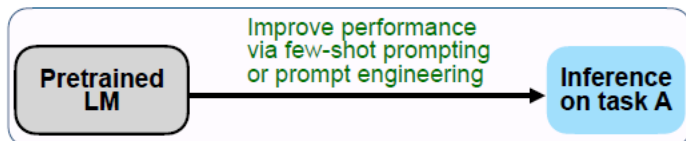
Inference on unseen task type



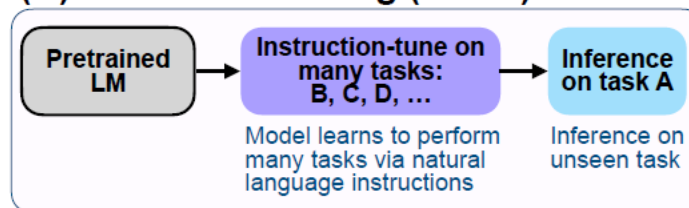
(A) Pretrain–finetune (BERT, T5)



(B) Prompting (GPT-3)



(C) Instruction tuning (FLAN)



FLAN: датасеты

- 62 датасета:

Natural language inference (7 datasets)

ANLI (R1-R3)	RTE
CB	SNLI
MNLI	WNLI
QNLI	

Commonsense (4 datasets)

CoPA
HellaSwag
PiQA
StoryCloze

Sentiment (4 datasets)

IMDB
Sent140
SST-2
Yelp

Paraphrase (4 datasets)

MRPC
QQP
PAWS
STS-B

Closed-book QA (3 datasets)

ARC (easy/chal.)
NQ
TQA

Struct to text (4 datasets)

CommonGen
DART
E2ENLG
WEBNLG

Translation (8 datasets)

ParaCrawl EN/DE
ParaCrawl EN/ES
ParaCrawl EN/FR
WMT-16 EN/CS
WMT-16 EN/DE
WMT-16 EN/FI
WMT-16 EN/RO
WMT-16 EN/RU
WMT-16 EN/TR

Reading comp. (5 datasets)

BoolQ	OBQA
DROP	SQuAD
MultiRC	

Read. comp. w/ commonsense (2 datasets)

CosmosQA
ReCoRD

Coreference (3 datasets)

DPR
Winogrande
WSC273

Misc. (7 datasets)

CoQA	TREC
QuAC	CoLA
WIC	Math
Fix Punctuation (NLG)	

Summarization (11 datasets)

AESLC	Multi-News	SamSum
AG News	Newsroom	Wiki Lingua EN
CNN-DM	Opin-Abs: iDebate	XSum
Gigaword	Opin-Abs: Movie	

FLAN: шаблоны инструкций

- 10 шаблонов на датасет

Premise

Russian cosmonaut Valery Polyakov set the record for the longest continuous amount of time spent in space, a staggering 438 days, between 1994 and 1995.

Hypothesis

Russians hold the record for the longest stay in space.

Target

Entailment
Not entailment



Options:
- yes
- no



Template 1

<premise>

Based on the paragraph above, can we conclude that
<hypothesis>?

<options>

Template 2

<premise>

Can we infer the following?

<hypothesis>

<options>

Template 3

Read the following and determine if the hypothesis can be inferred from the premise:

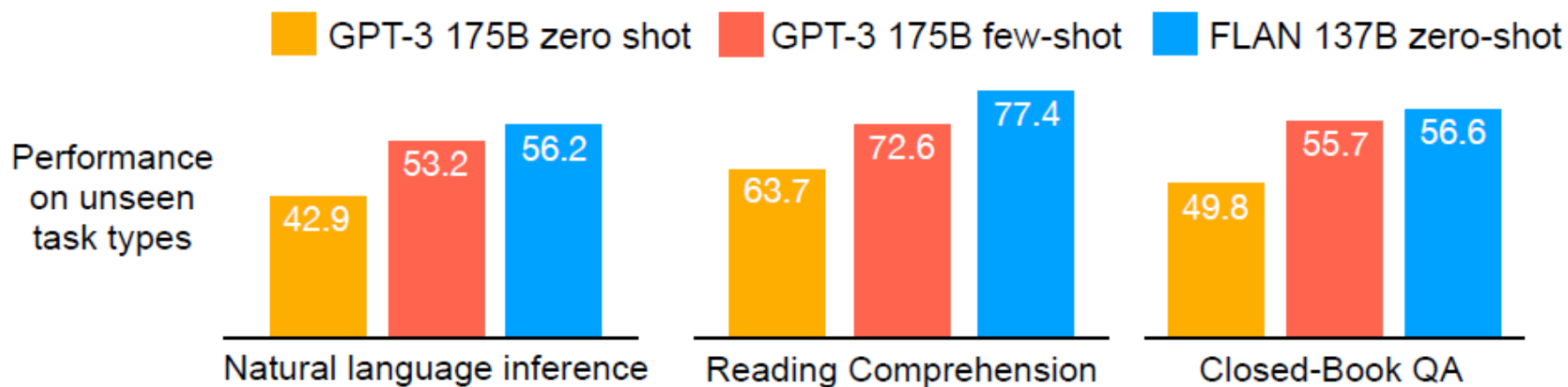
Premise: <premise>

Hypothesis: <hypothesis>

<options>

Template 4, ...

FLAN: результаты



RETRO

- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford et al. Improving language models by retrieving from trillions of tokens // arXiv:2112.04426
 - RETRO = Retrieval-Enhanced Transformer
 - DeepMind (декабрь 2021)
- Основная идея: включение поиска на основе к ближайших соседей в Transformer

RETRO: архитектура

