Языковые модели

Майоров Владимир

27 октября 2021

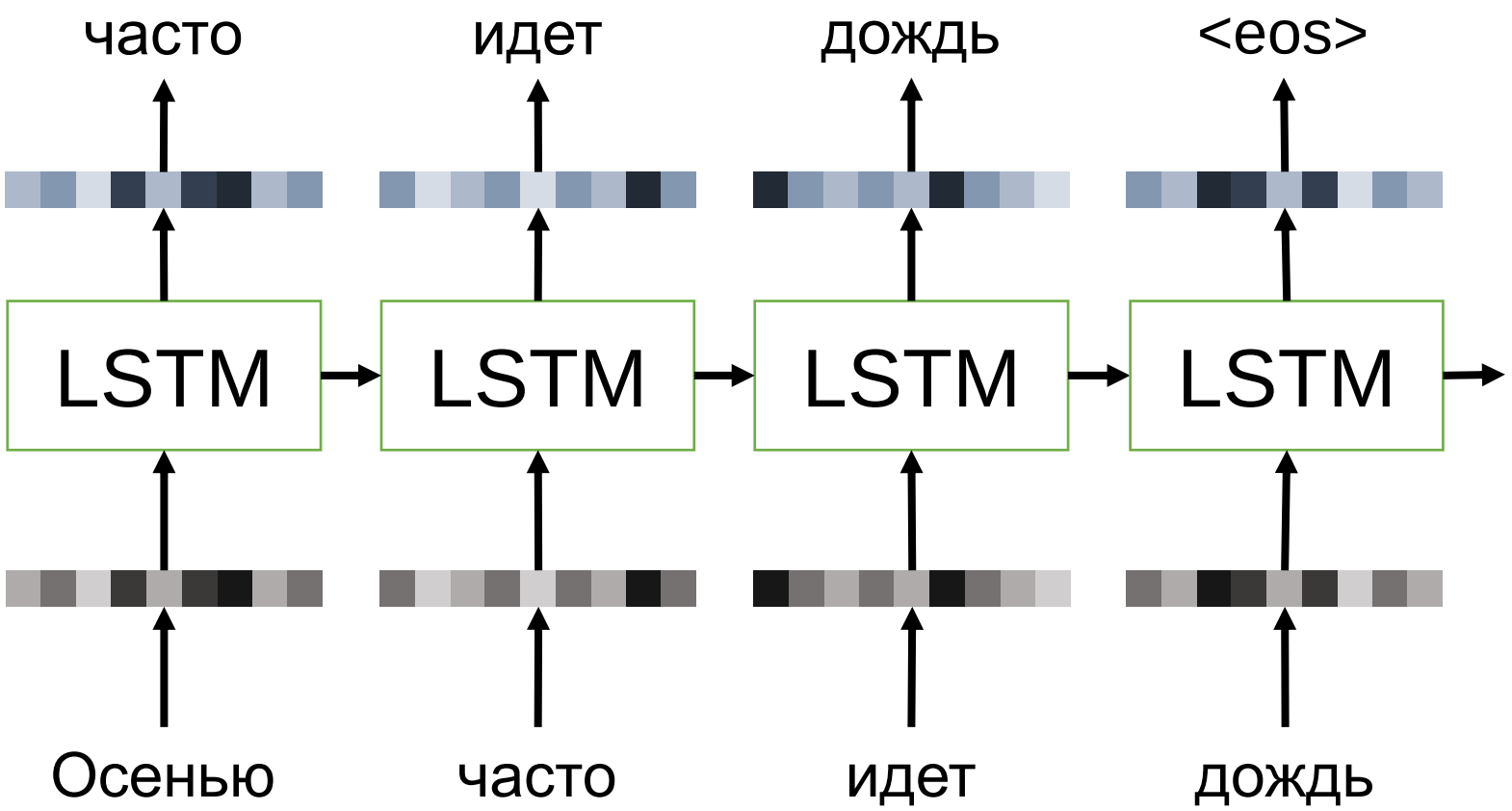
ИСП РАН

Задача языкового моделирования

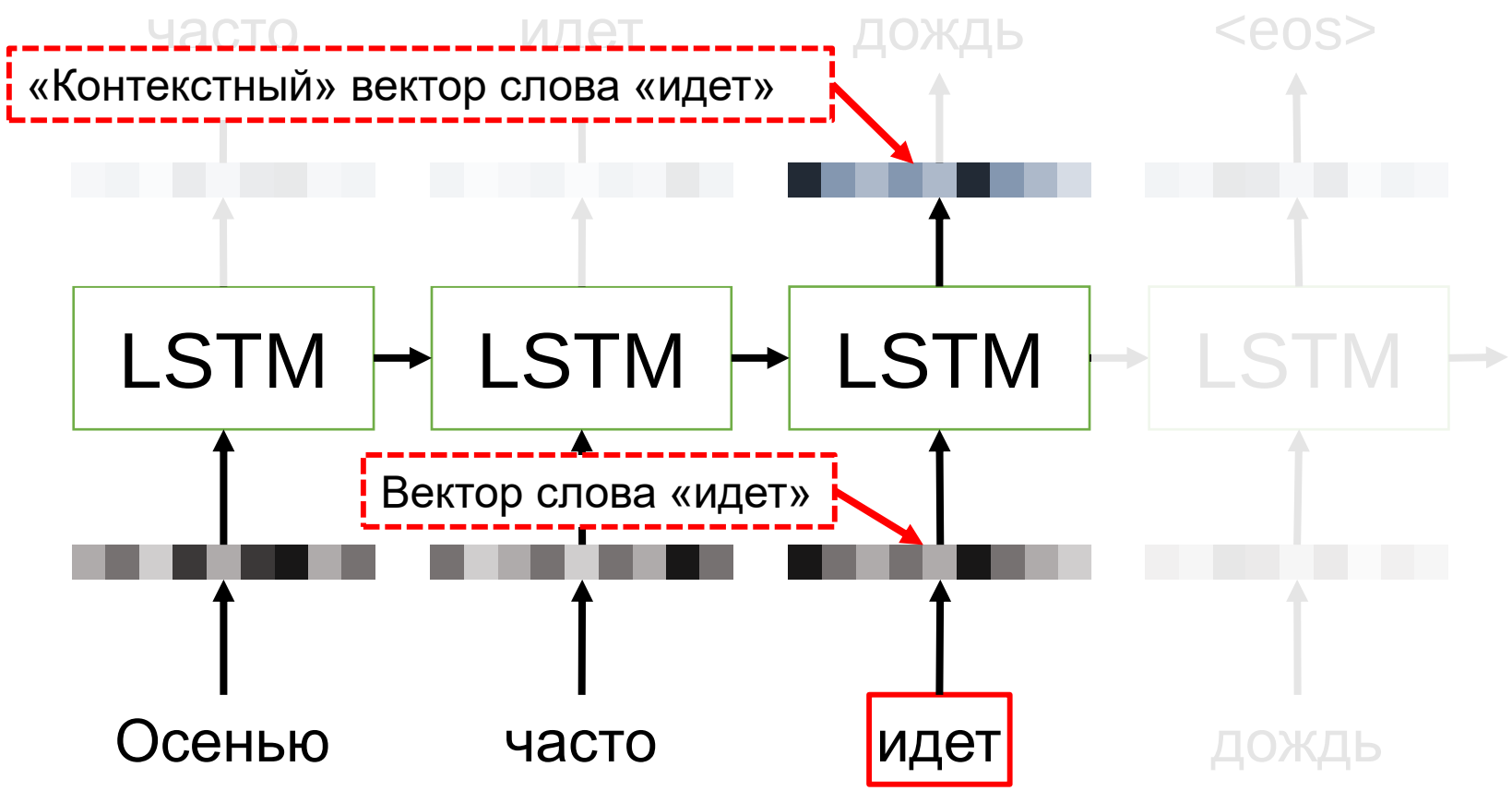
Языковая модель – это вероятностное распределение на множестве последовательностей слов:

- $P(w_1, w_2, \ldots, w_n)$
  - $P(\text{из окна сильно дуло}) = 0.0001$
  - $P(\text{дуло окна из сильно}) = 0.0000000000001$
- $P(w_n | w_1, \ldots, w_{n-1})$
  - Осенью часто идет …
  - Село Коровка в качестве …

Языковые модели на RNN



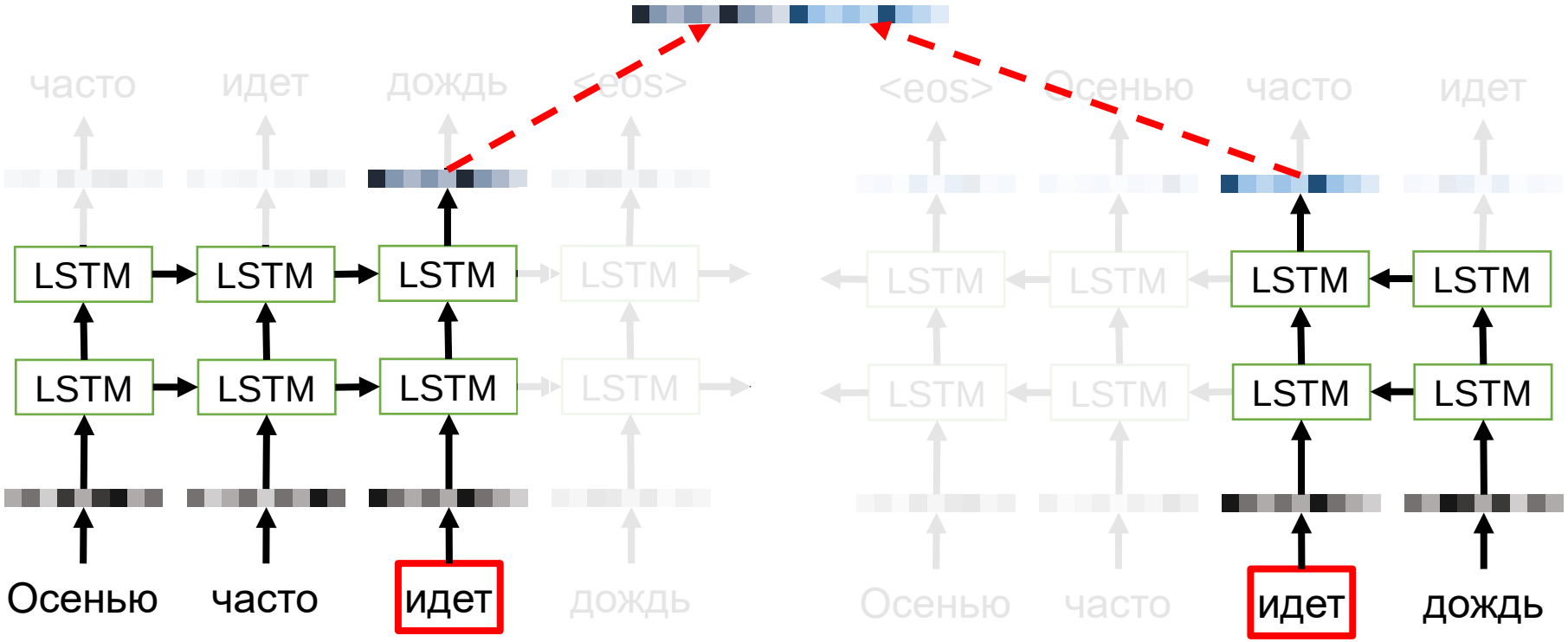
Языковые модели на RNN



Language Model Embedding

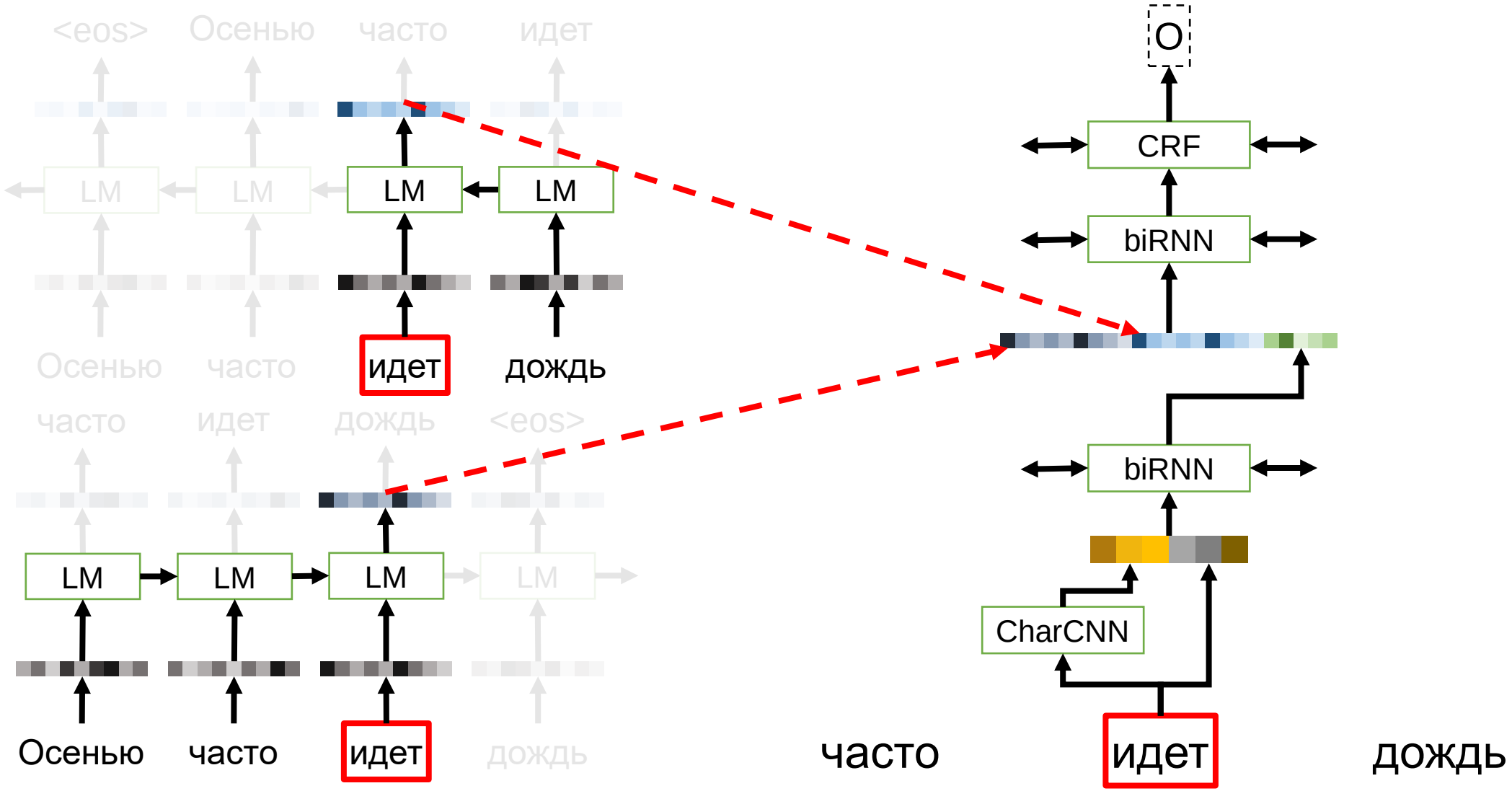
ИСП РАН

- Прямая модель: $p(w_1, \ldots, w_N) = \prod_{k=1}^{N} p(w_k | w_1, \ldots, w_{k-1})$
- Обратная модель: $p(w_1, \ldots, w_N) = \prod_{k=1}^{N} p(w_k | w_{k+1}, \ldots, w_N)$



Peters M. E. et al. Semi-supervised sequence tagging with bidirectional language models – 2017.

Language Model Embedding



ELMo (Embeddings from Language Models)

ИСП РАН

- Прямая модель:

$$p(w_1, \dots, w_N) = \prod_{k=1}^{N} p(w_k | w_1, \dots, w_{k-1})$$

- Обратная модель:

$$p(w_1, \dots, w_N) = \prod_{k=1}^{N} p(w_k | w_{k+1}, \dots, w_N)$$

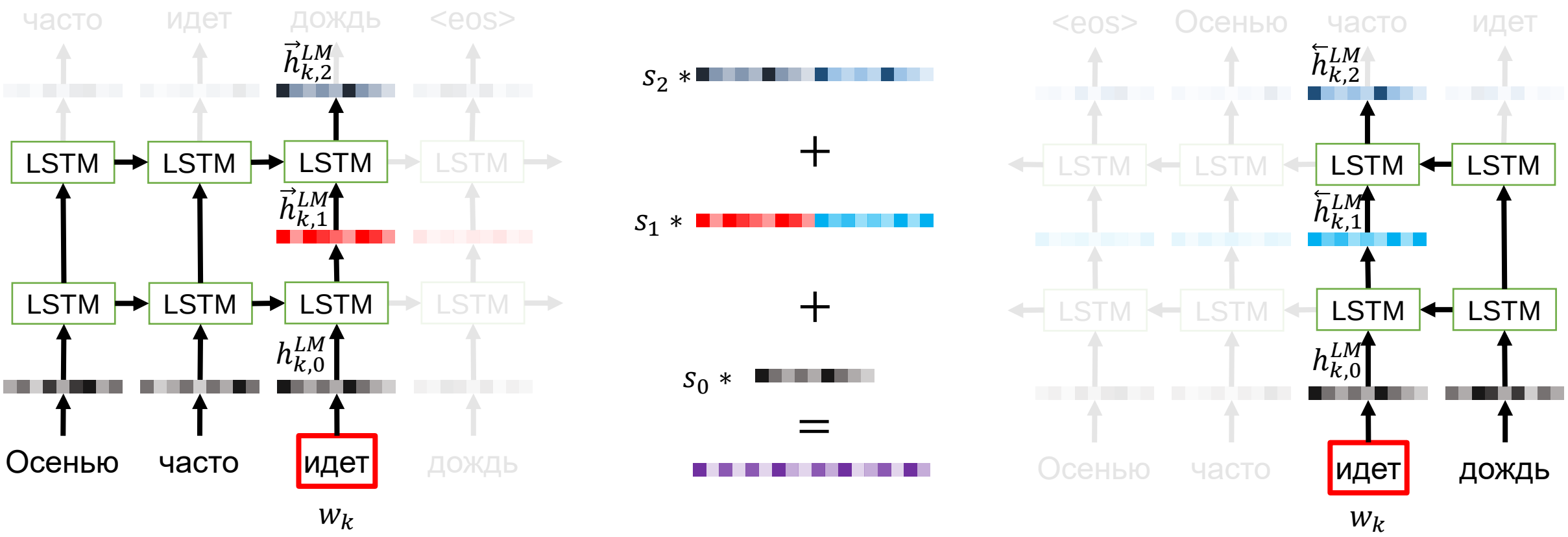
- Обучаем прямую и обратную языковую модель одновременно:

$$J = \sum_{k=1}^{N} [\log p(w_k | w_1, \dots, w_{k-1}; \Theta_x, \Theta_{\overrightarrow{LSTM}}, \Theta_s) + \log p(w_k | w_{k+1}, \dots, w_N; \Theta_x, \Theta_{\overleftarrow{LSTM}}, \Theta_s)]$$

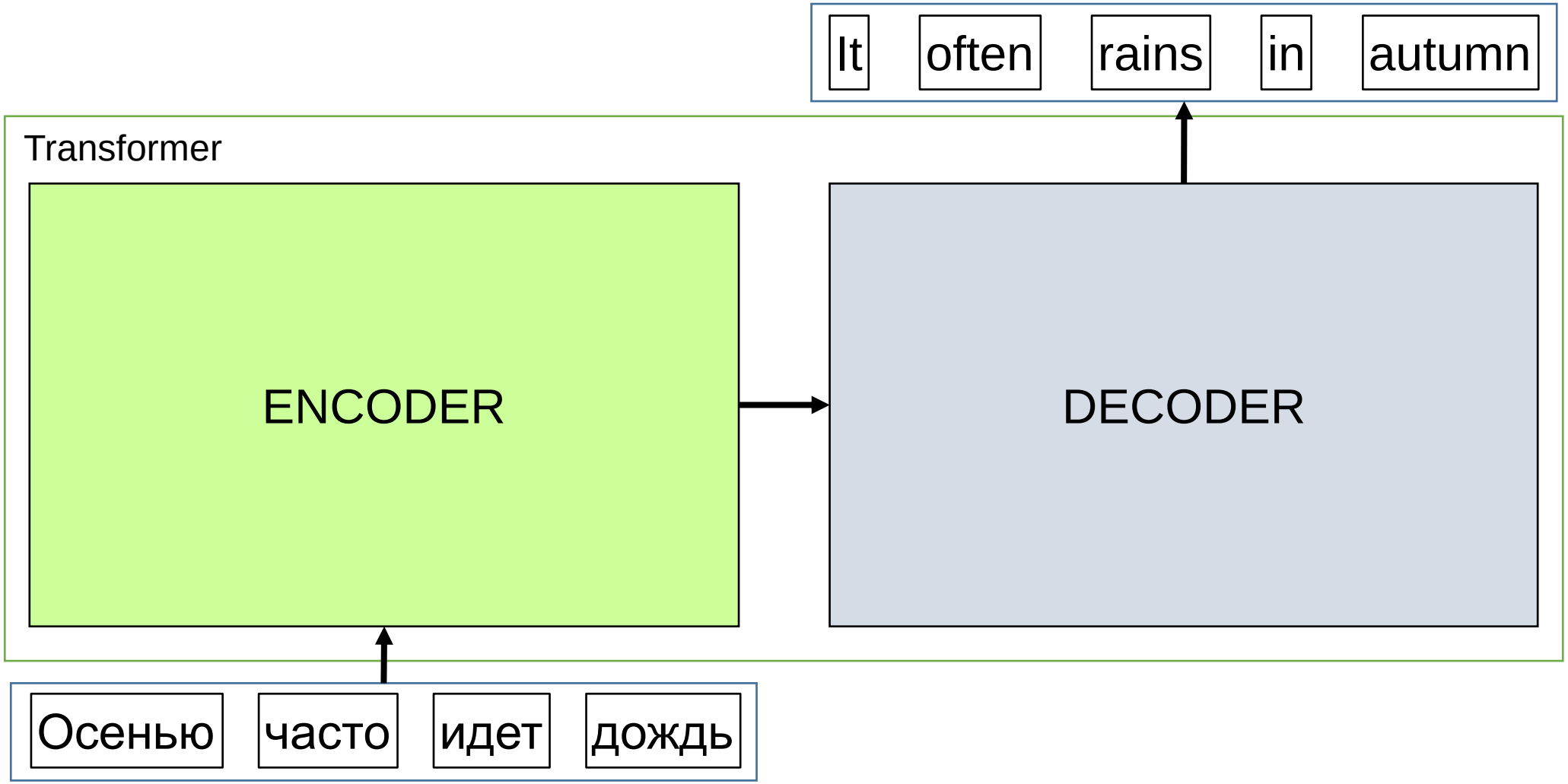
Peters M. E. et al. Deep contextualized word representations – 2018.

ELMo (Embeddings from Language Models)

$$ELMo_k = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}, \text{где } h_{k,i}^{LM} = [\overrightarrow{h}_{k,i}^{LM}; \overleftarrow{h}_{k,i}^{LM}], i = \overline{1,n}$$



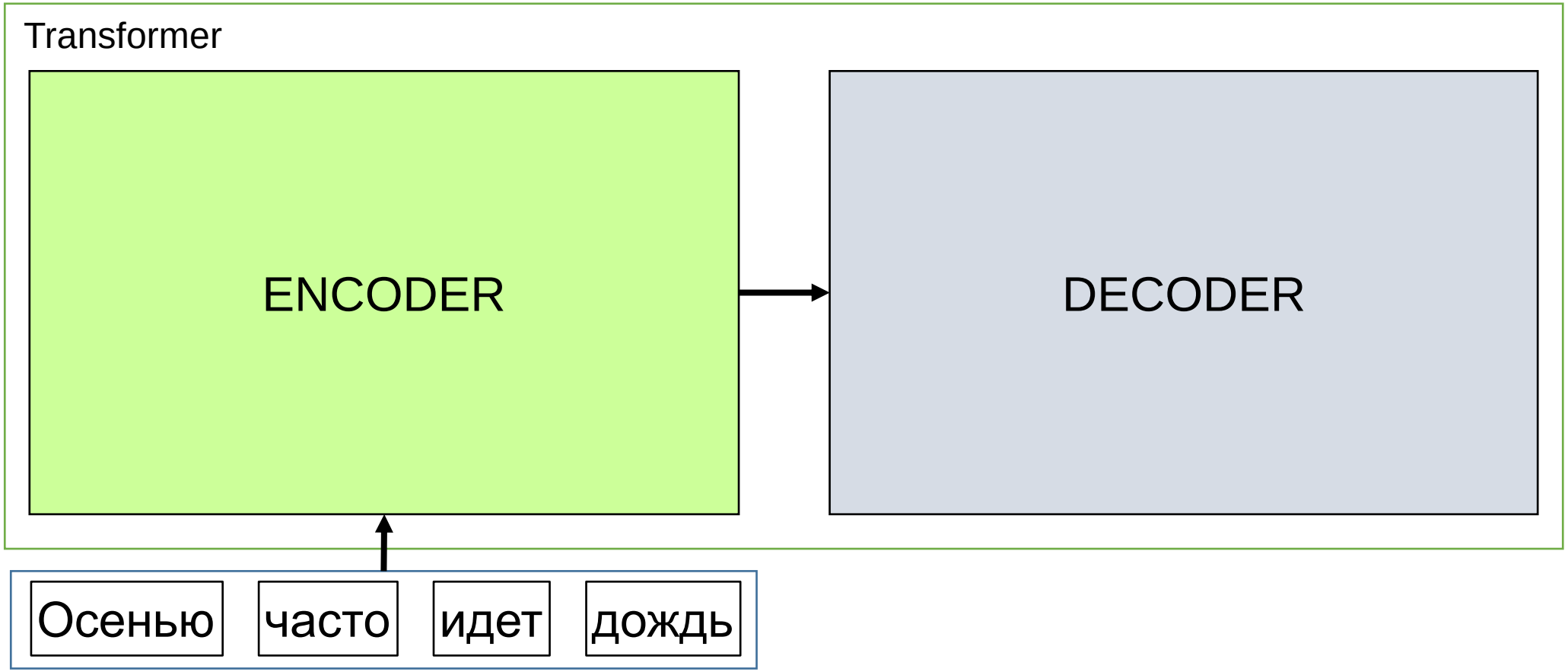
Transformer*



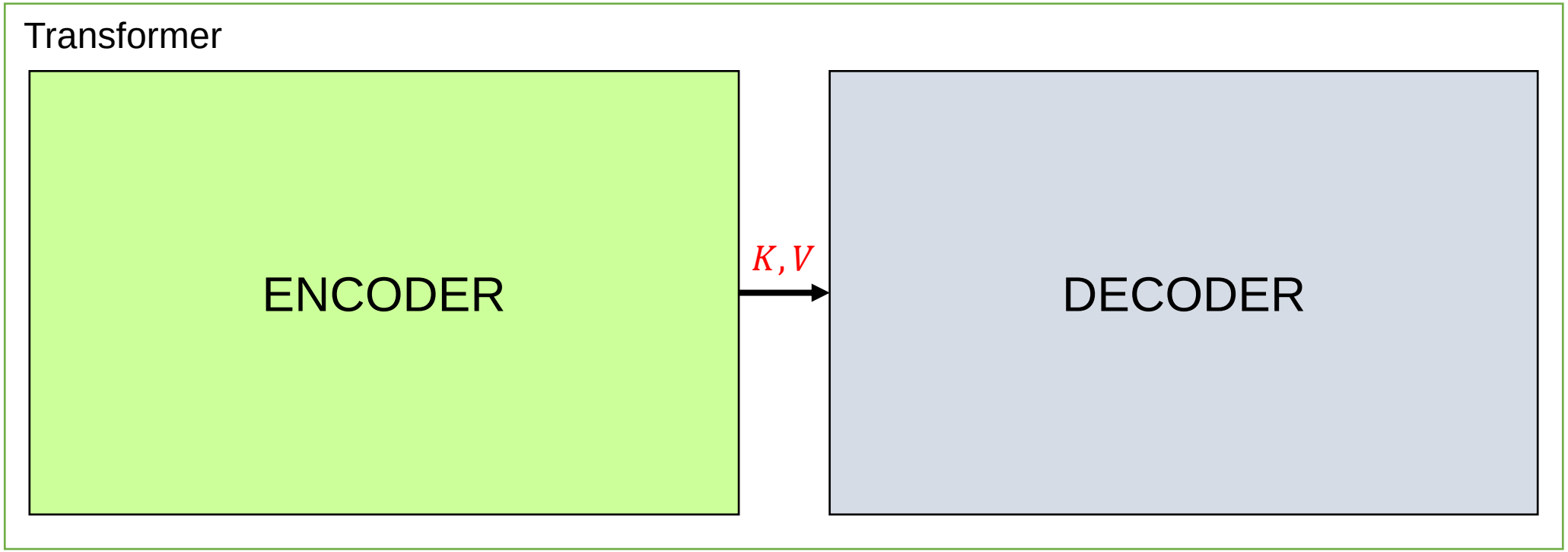
*подробное объяснение про трансформер есть здесь

Vaswani A. et al. Attention is all you need – 2017

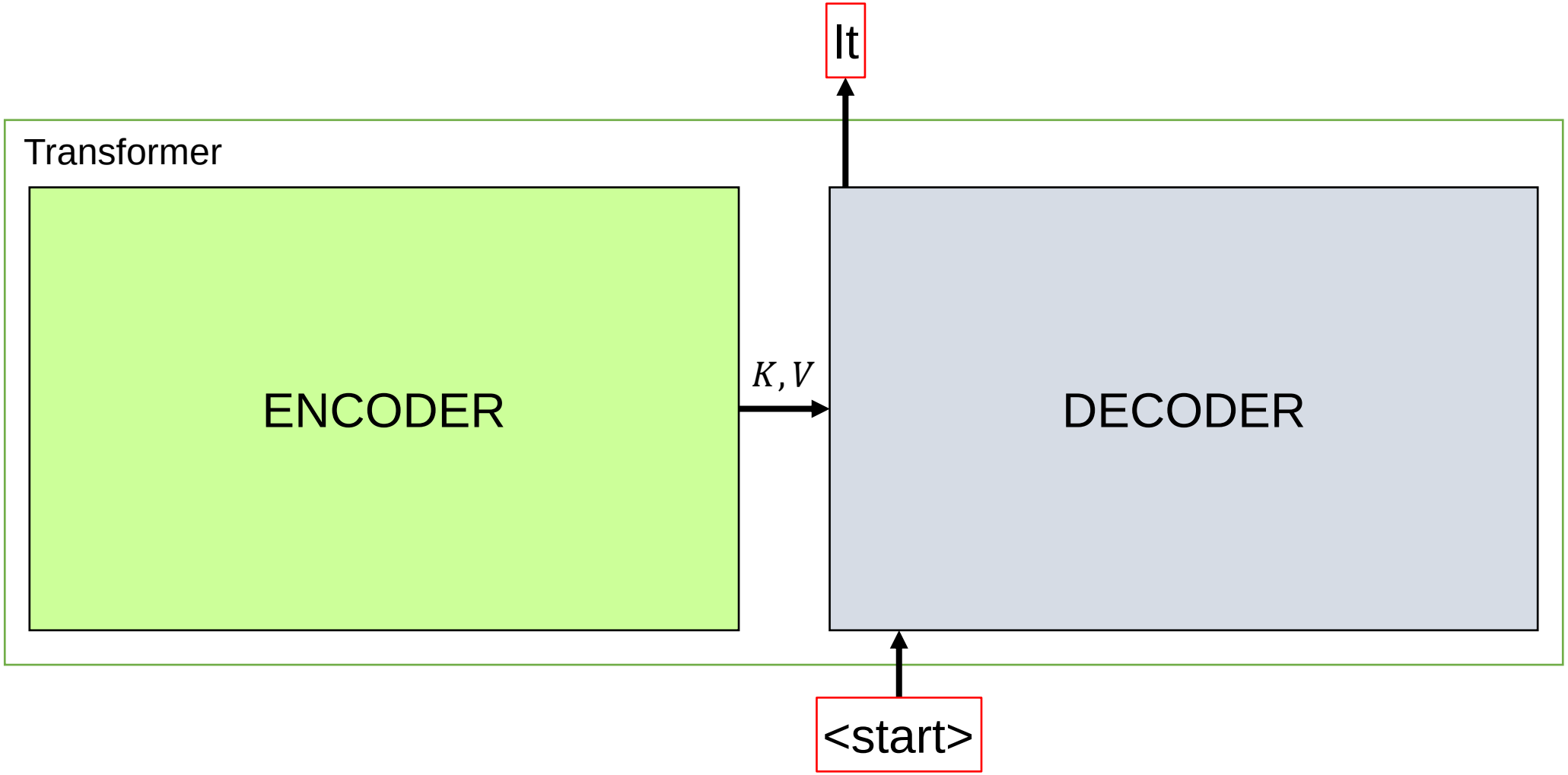
Transformer*



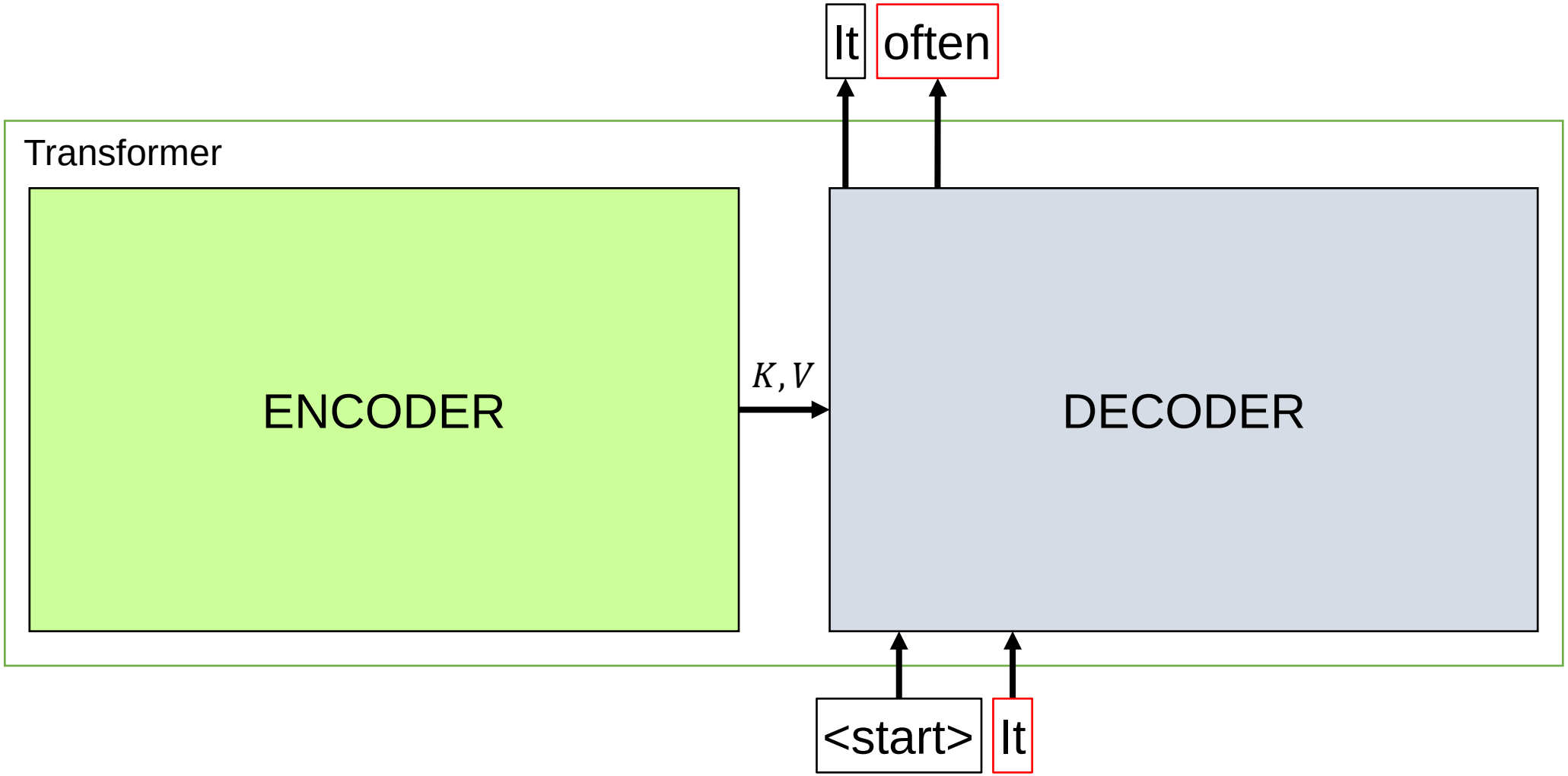
Transformer*



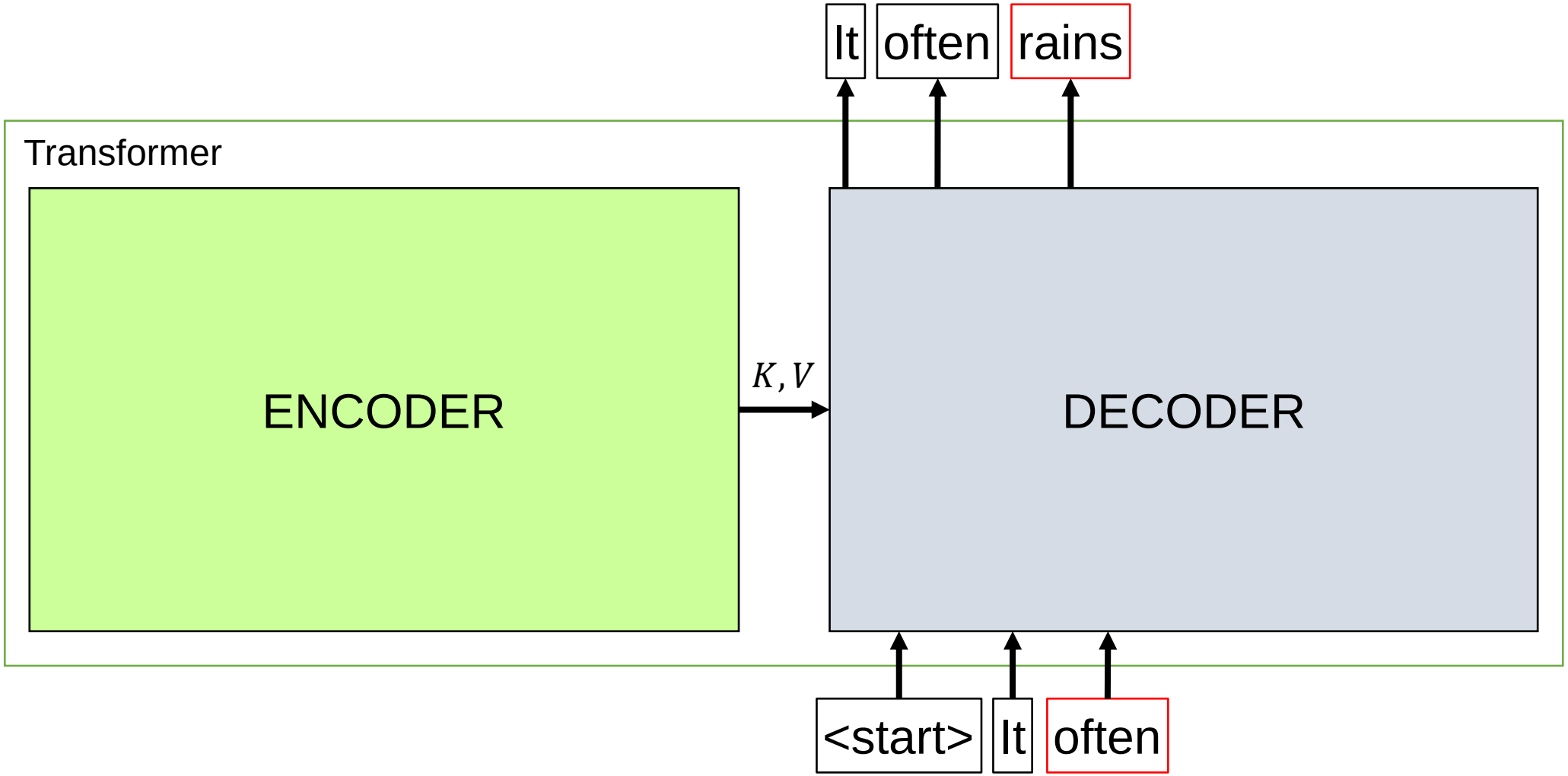
Transformer*



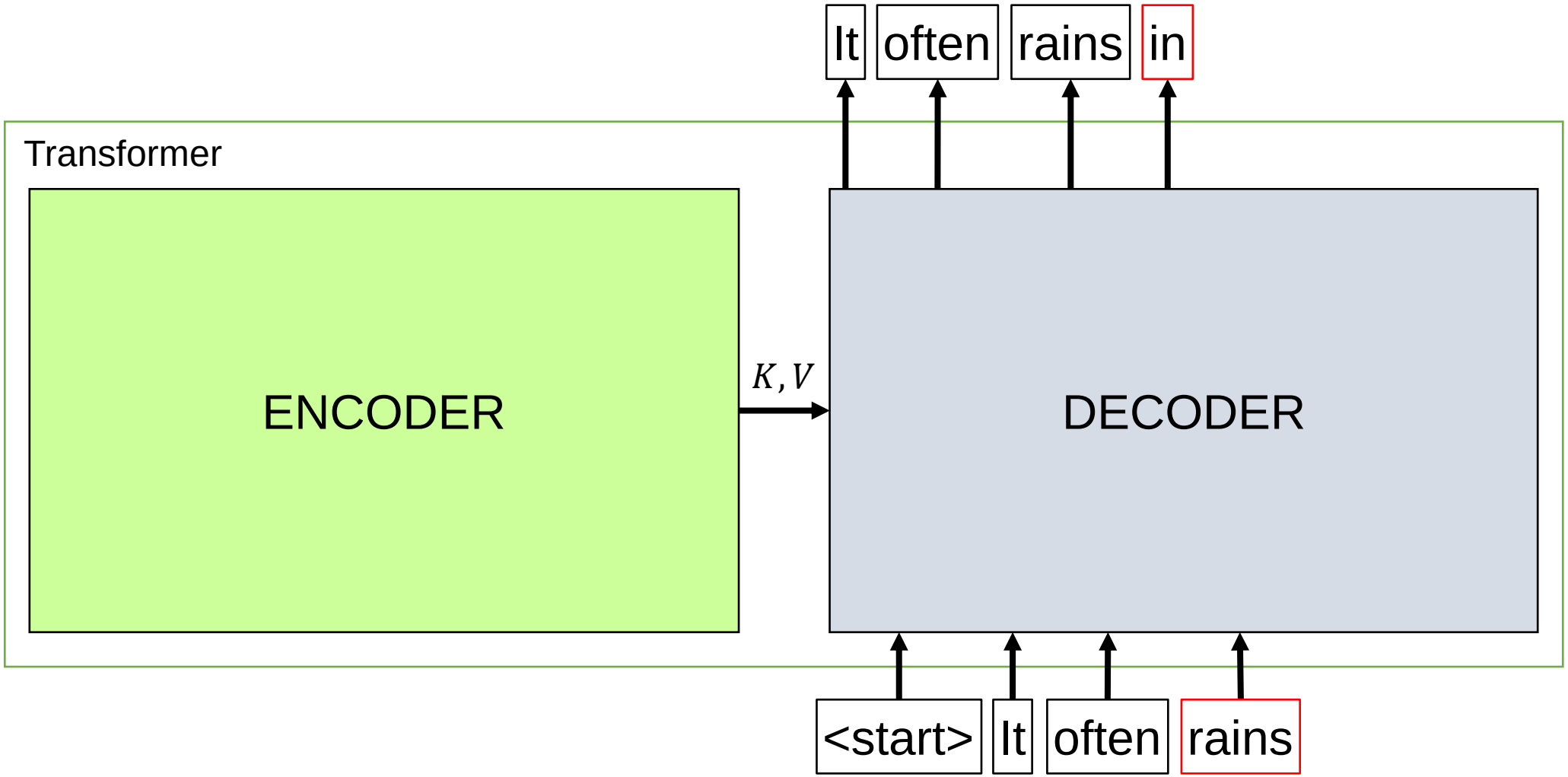
Transformer*



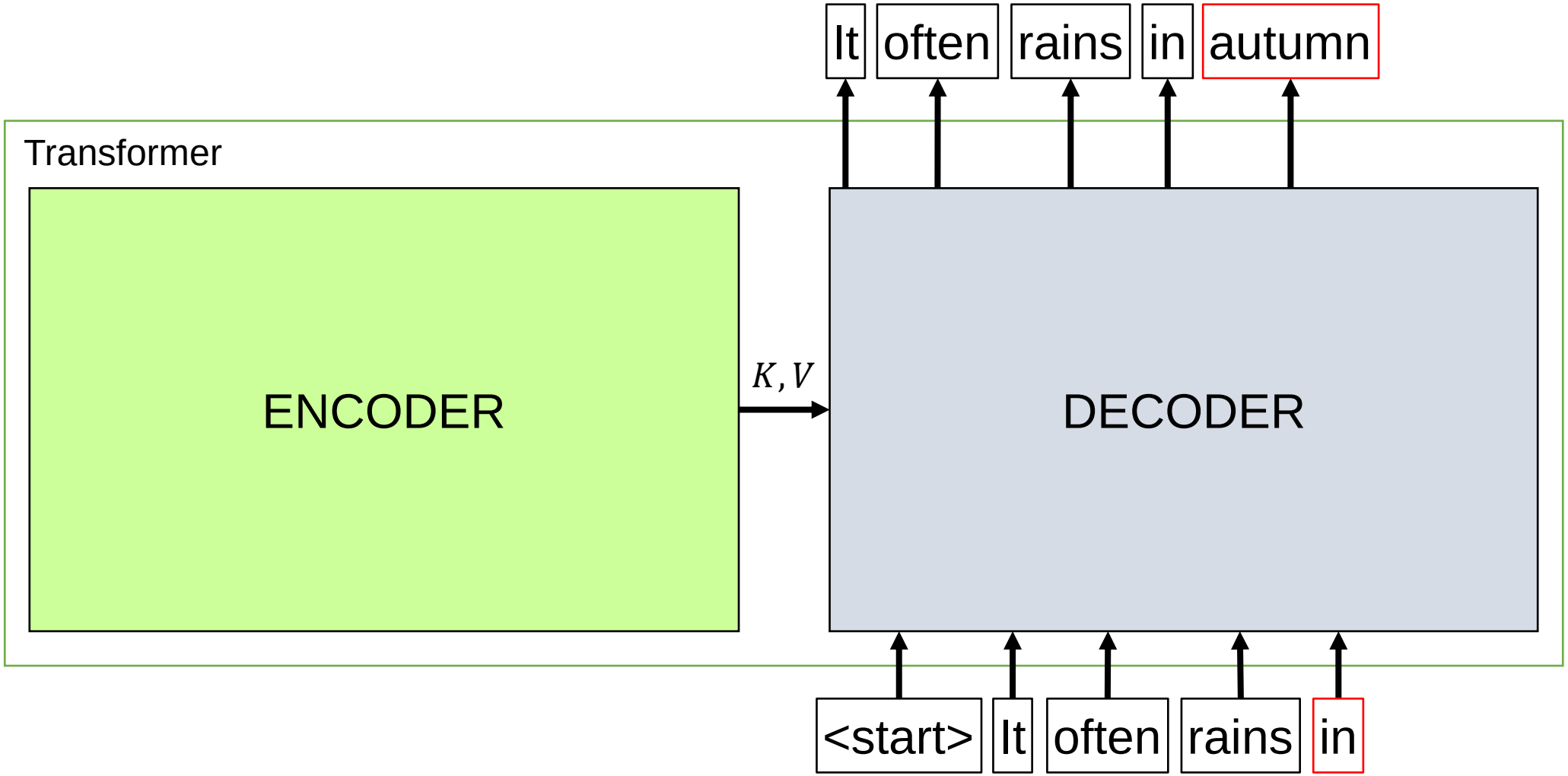
Transformer*



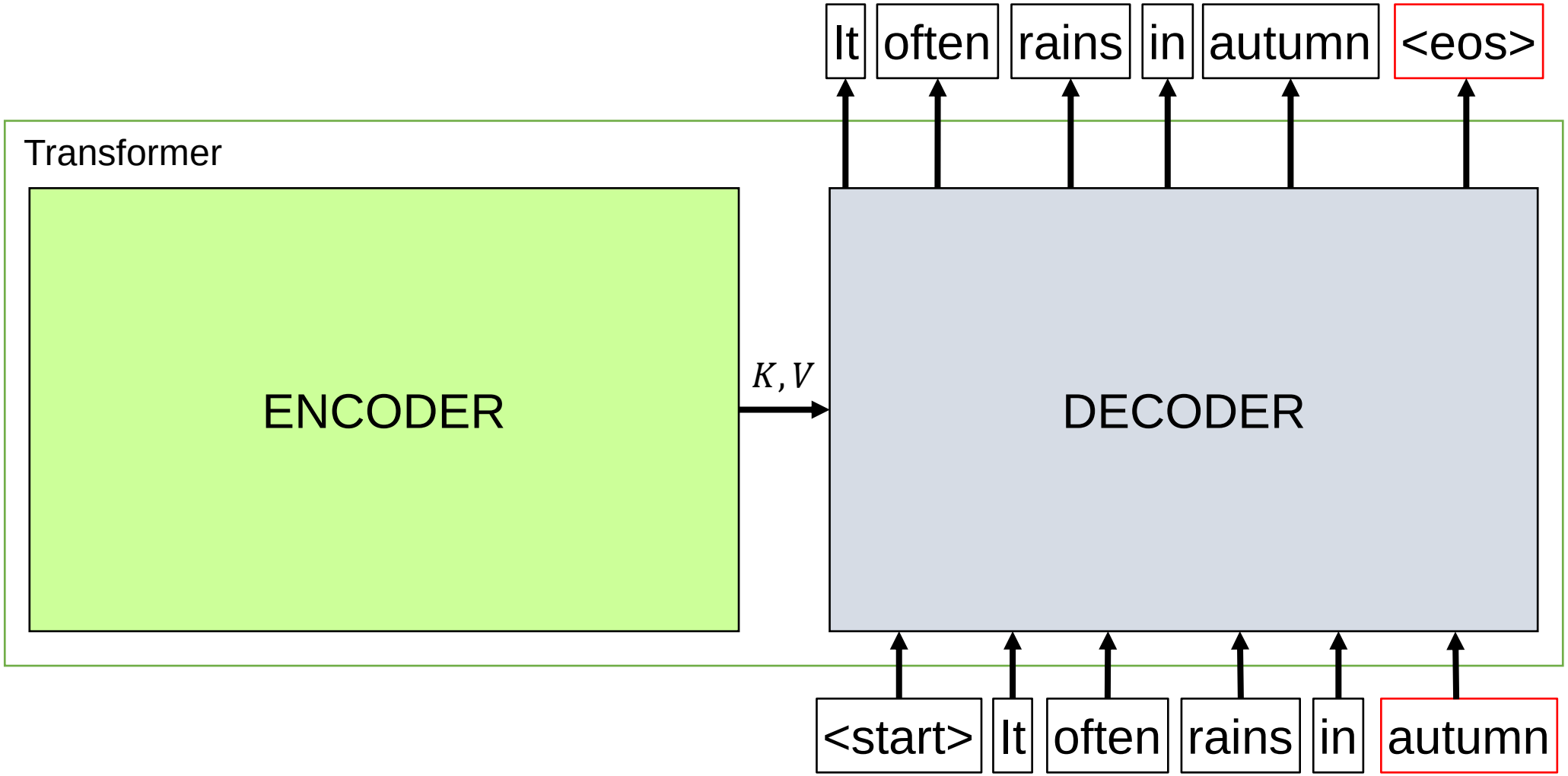
Transformer*



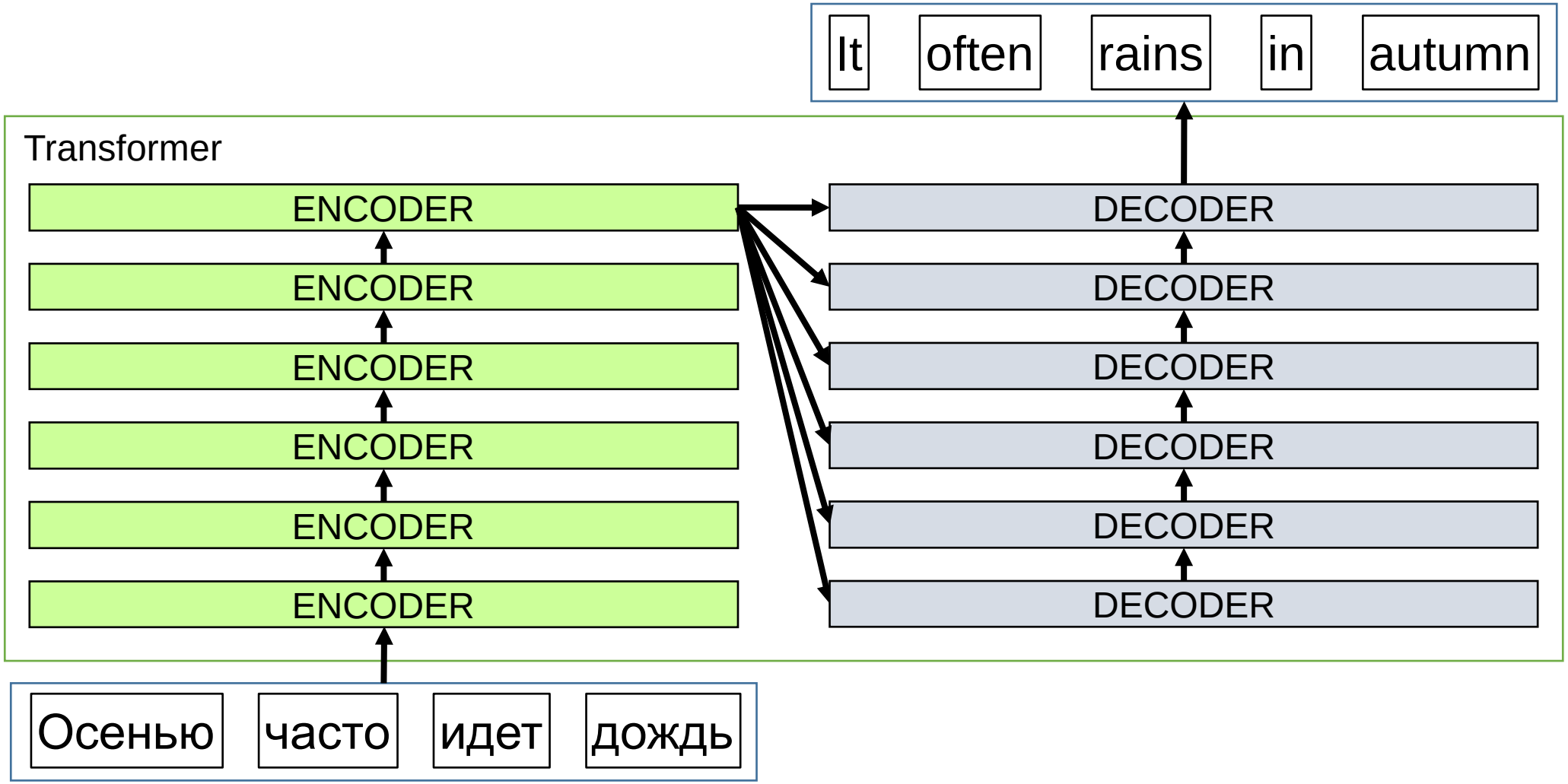
Transformer*



Transformer*



Transformer*



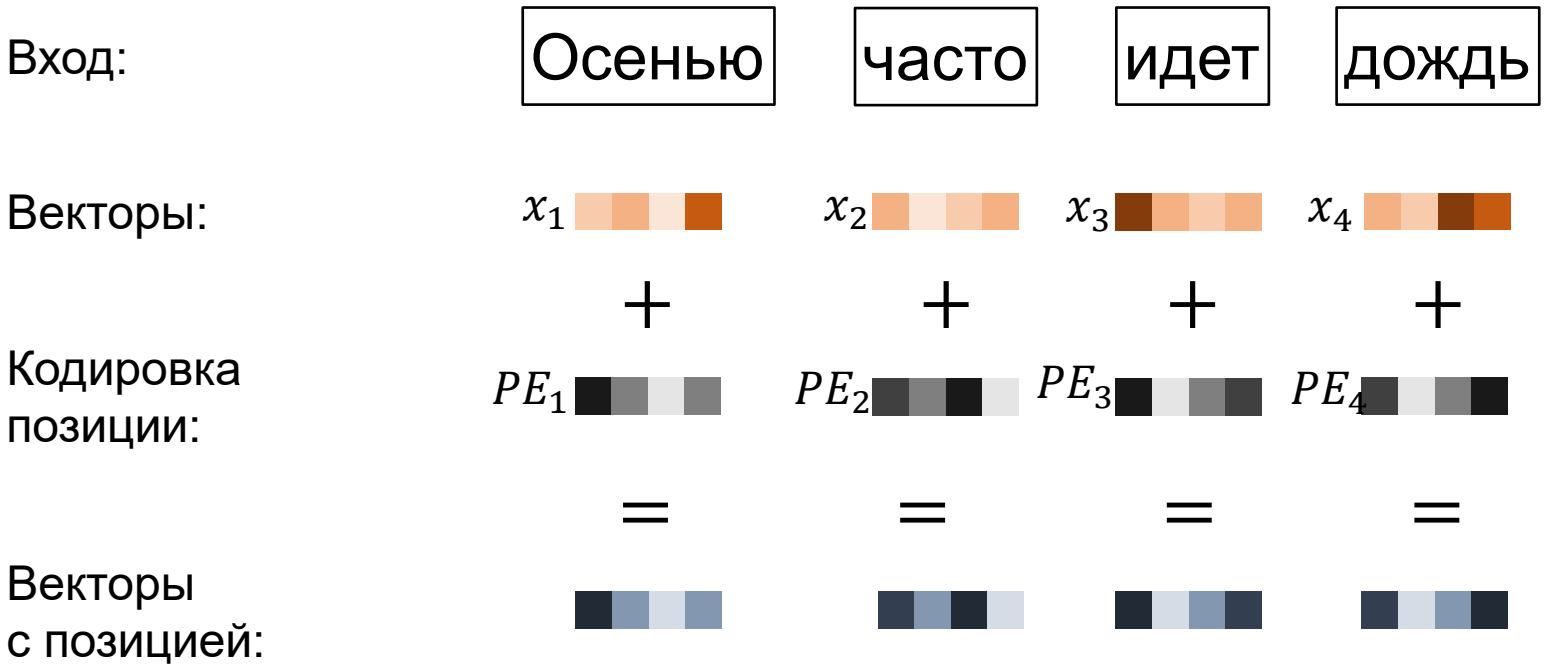
BPE (Byte Pair Encoding)

ИСП РАН

- Метод токенизации с фиксированным (небольшим размером словаря)
  1. Словарь инициализируется всеми символами из train (добавляется символ конца слова ∘)
  2. В цикле n раз:
     1. Выбрать самую частую пару символов (`A`, `B`)
     2. Добавить в словарь символ `AB`

Sennrich R., et al. Neural machine translation of rare words with subword units– 2015.

Позиционное кодирование



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

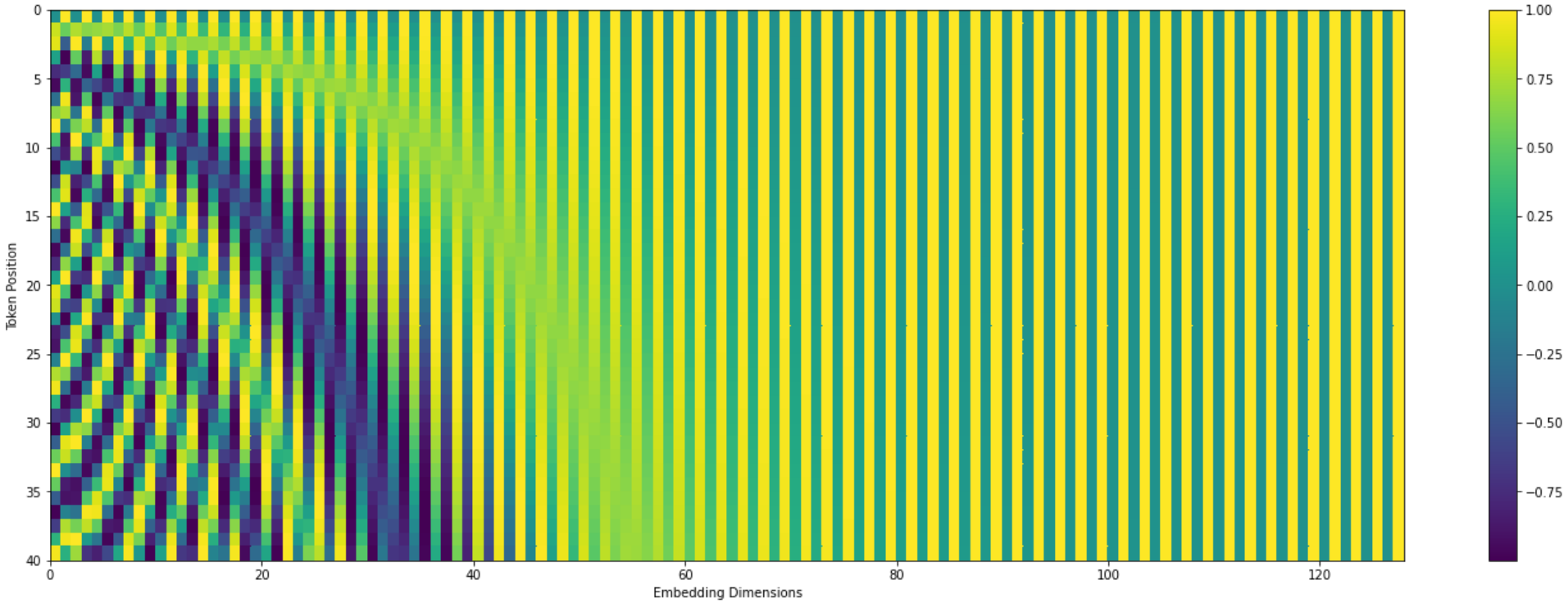
$$PE_{pos,2i+1} = \cos(^{pos}/_{10000^{2i/d_{model}}})$$

Позиционное кодирование

ИСП РАН

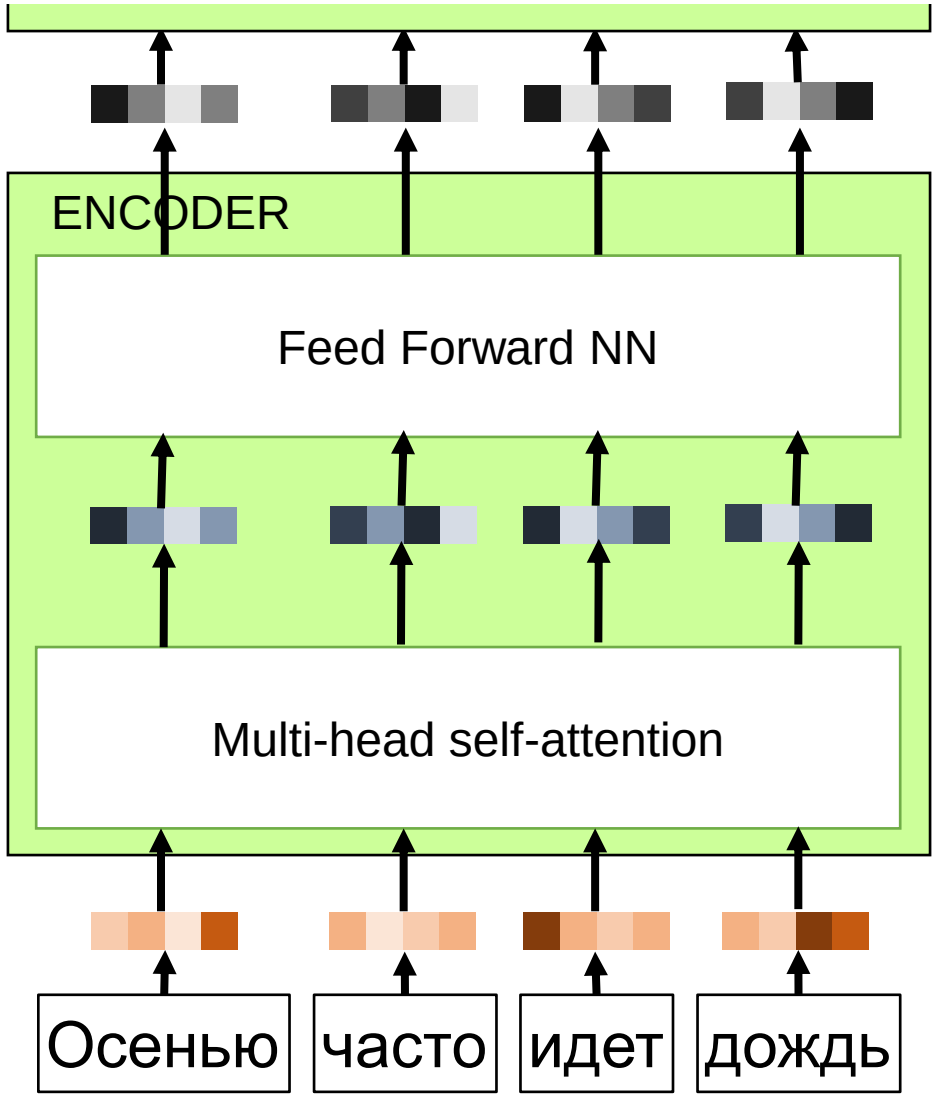
$$PE_{pos,2i} = \sin(^{pos}/_{10000^{2i/d_{model}}})$$

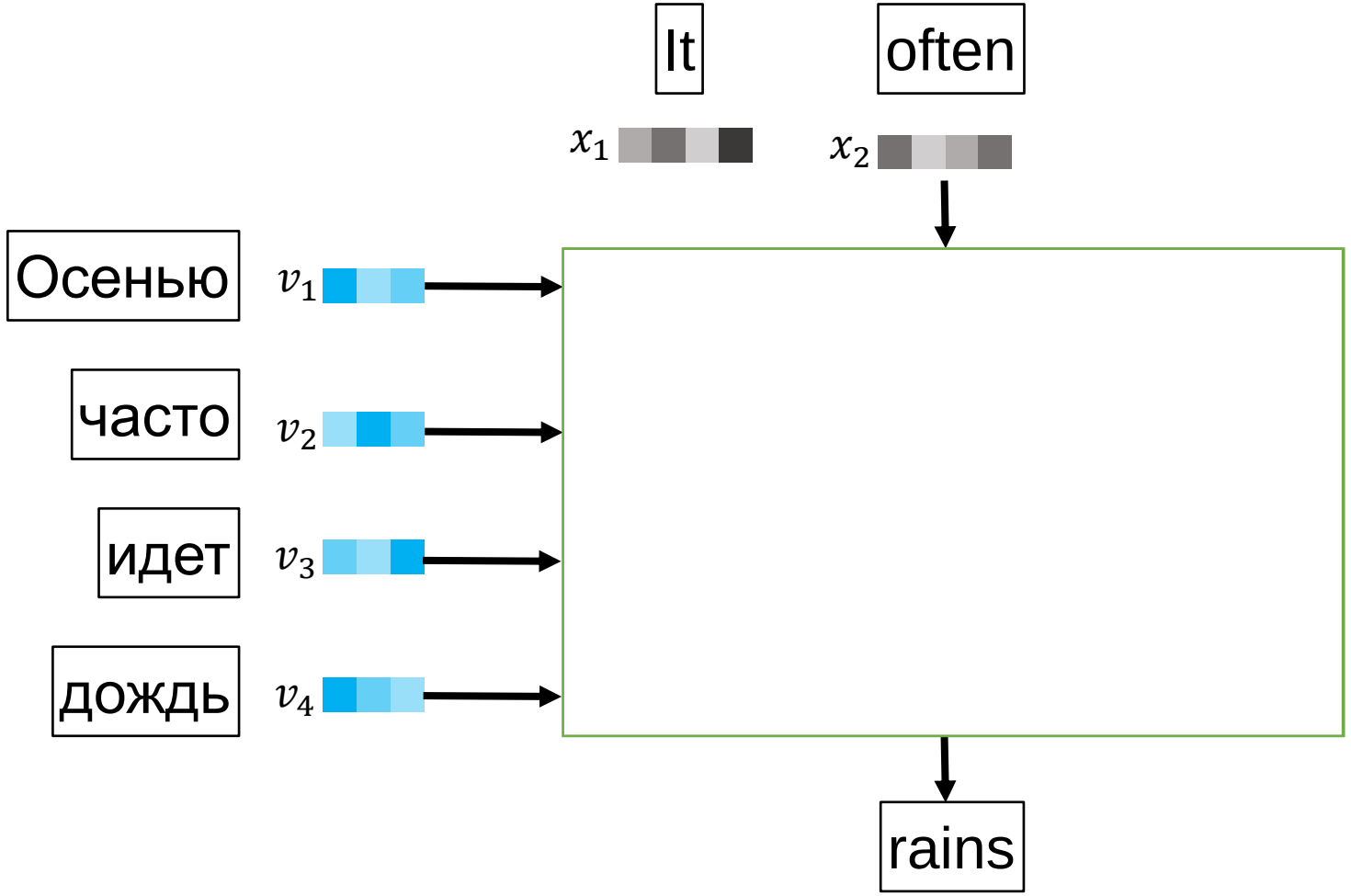
$$PE_{pos,2i+1} = \cos(^{pos}/_{10000^{2i/d_{model}}})$$

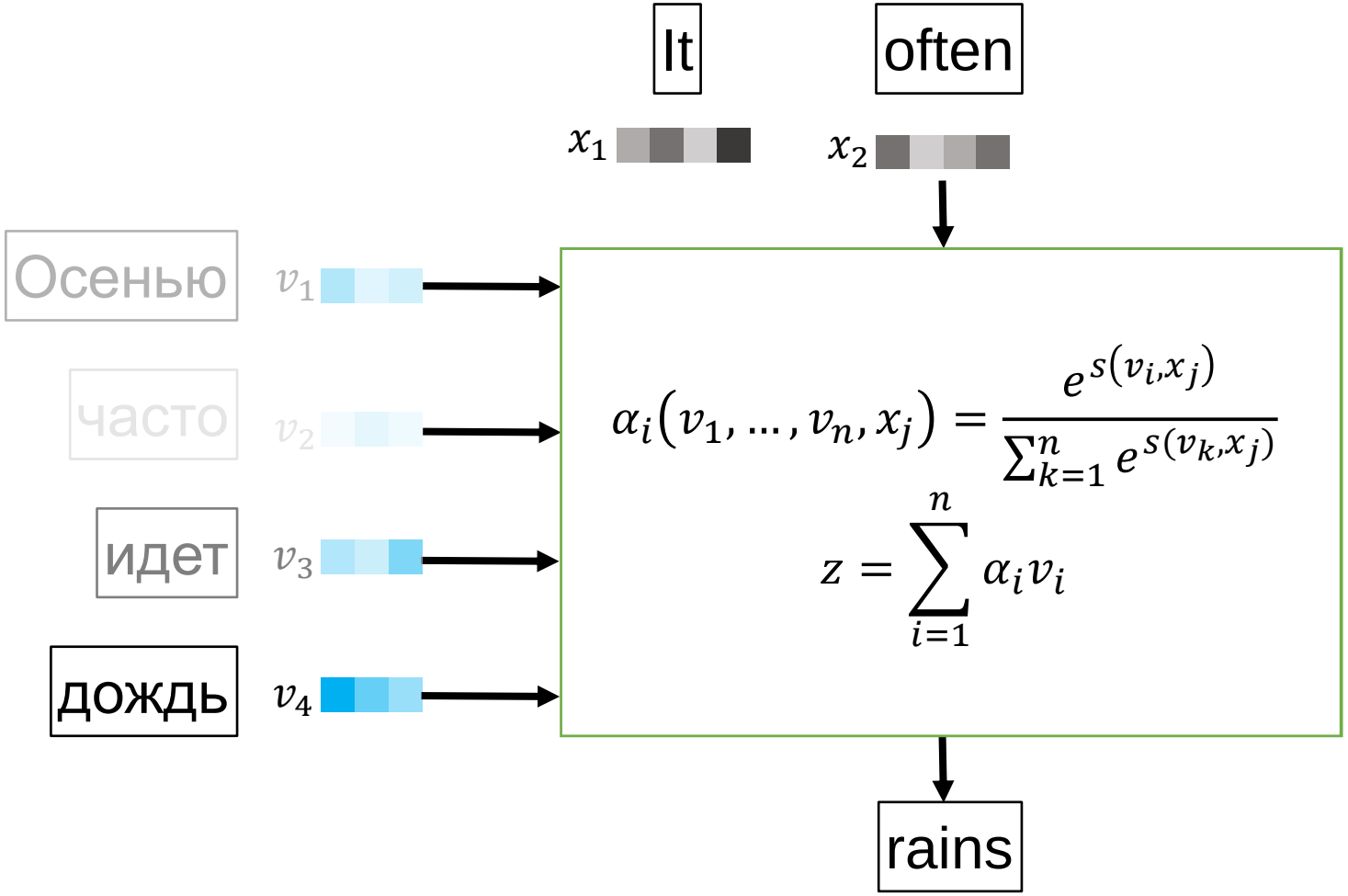


Код для генерации картинки взят здесь

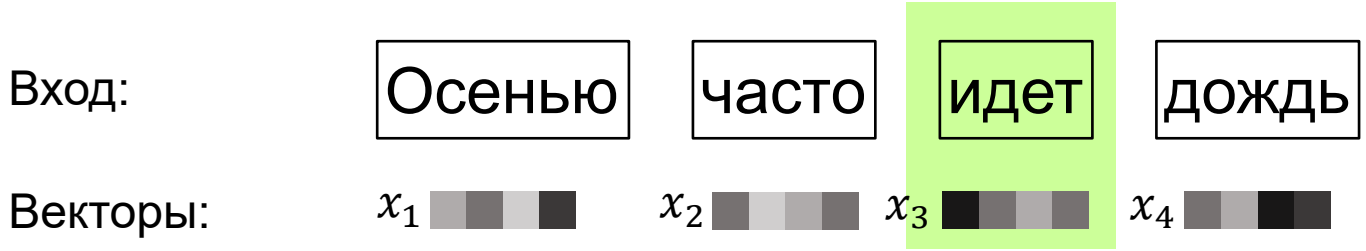
Transformer Encoder



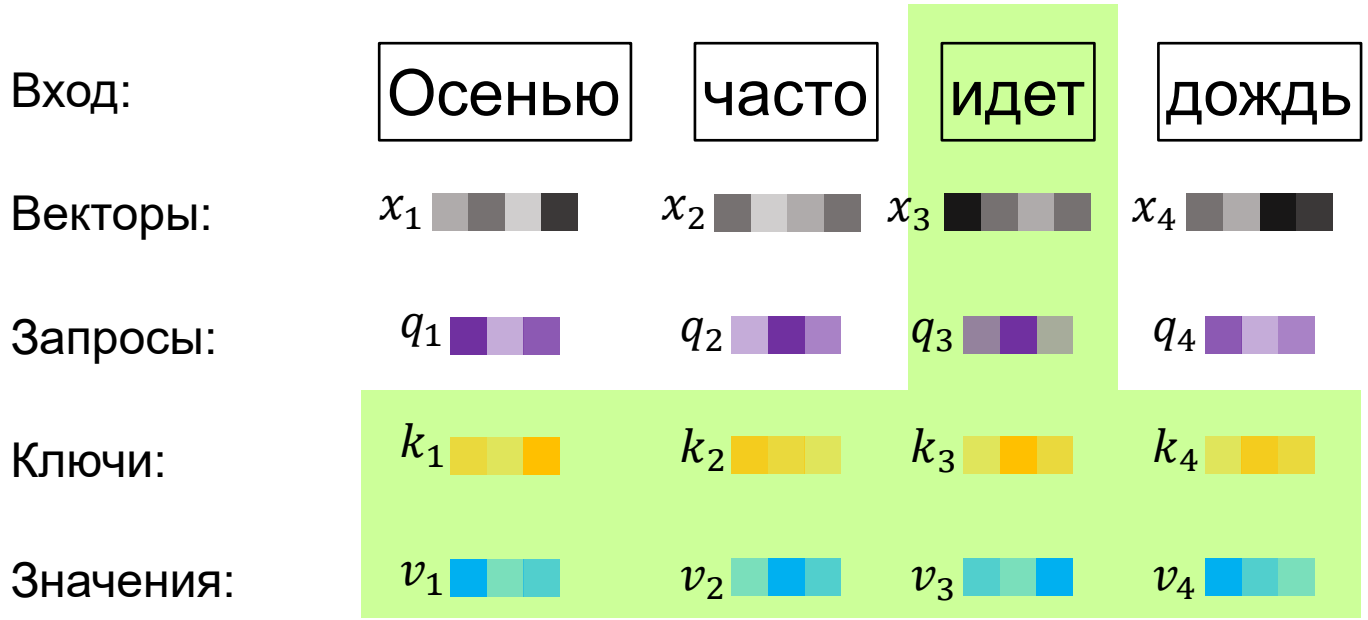

It
often
x_1
x_2
Осенью
часто
идет
дождь
v_1
v_2
v_3
v_4
rains


It
often
x_1
x_2
Осенью
часто
идет
дождь
v_1
v_2
v_3
v_4
$\alpha_i(v_1, \ldots, v_n, x_j) = \frac{e^{s(v_i, x_j)}}{\sum_{k=1}^{n} e^{s(v_k, x_j)}}$
$z = \sum_{i=1}^{n} \alpha_i v_i$
rains

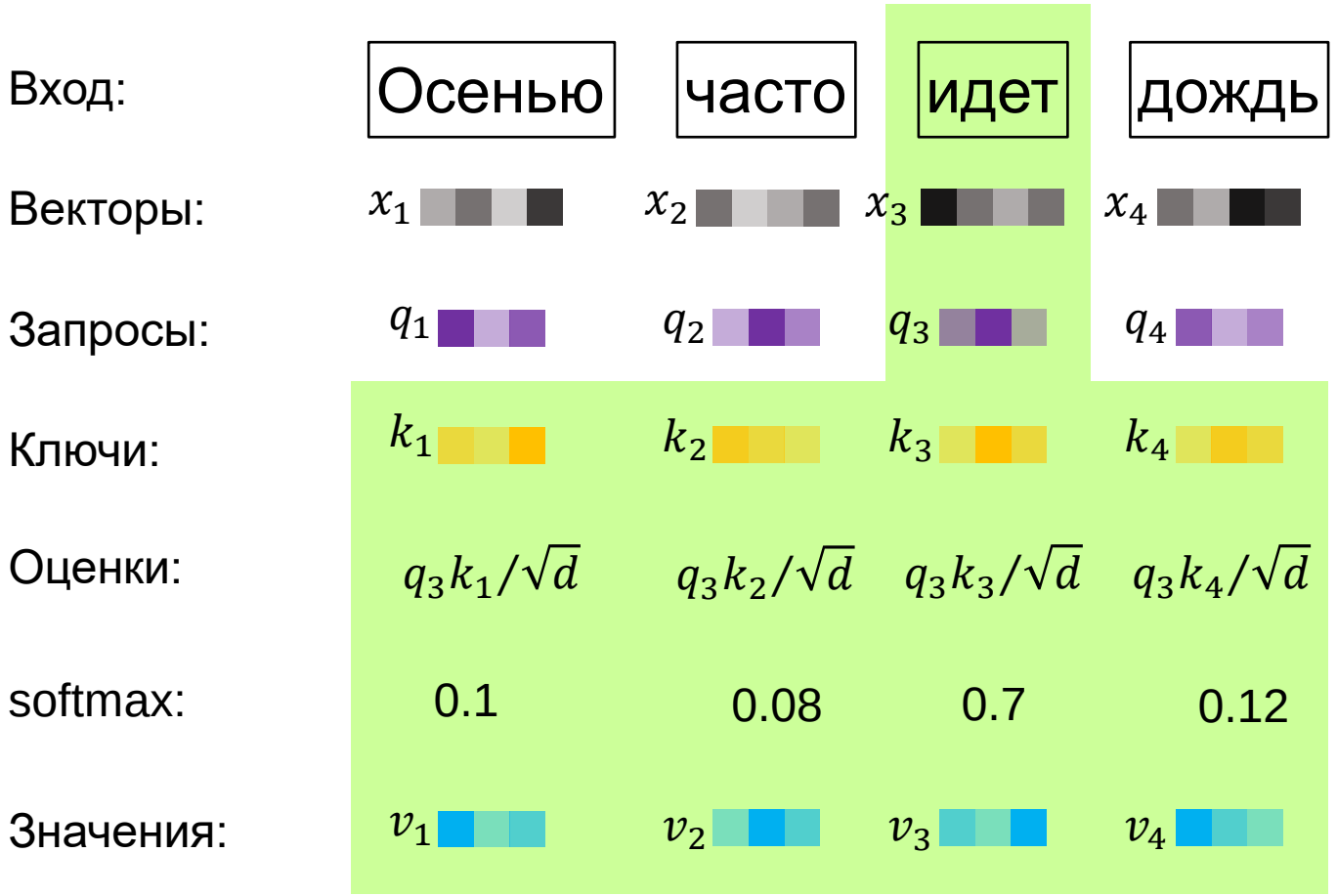
Scaled Dot-Product Self-attention



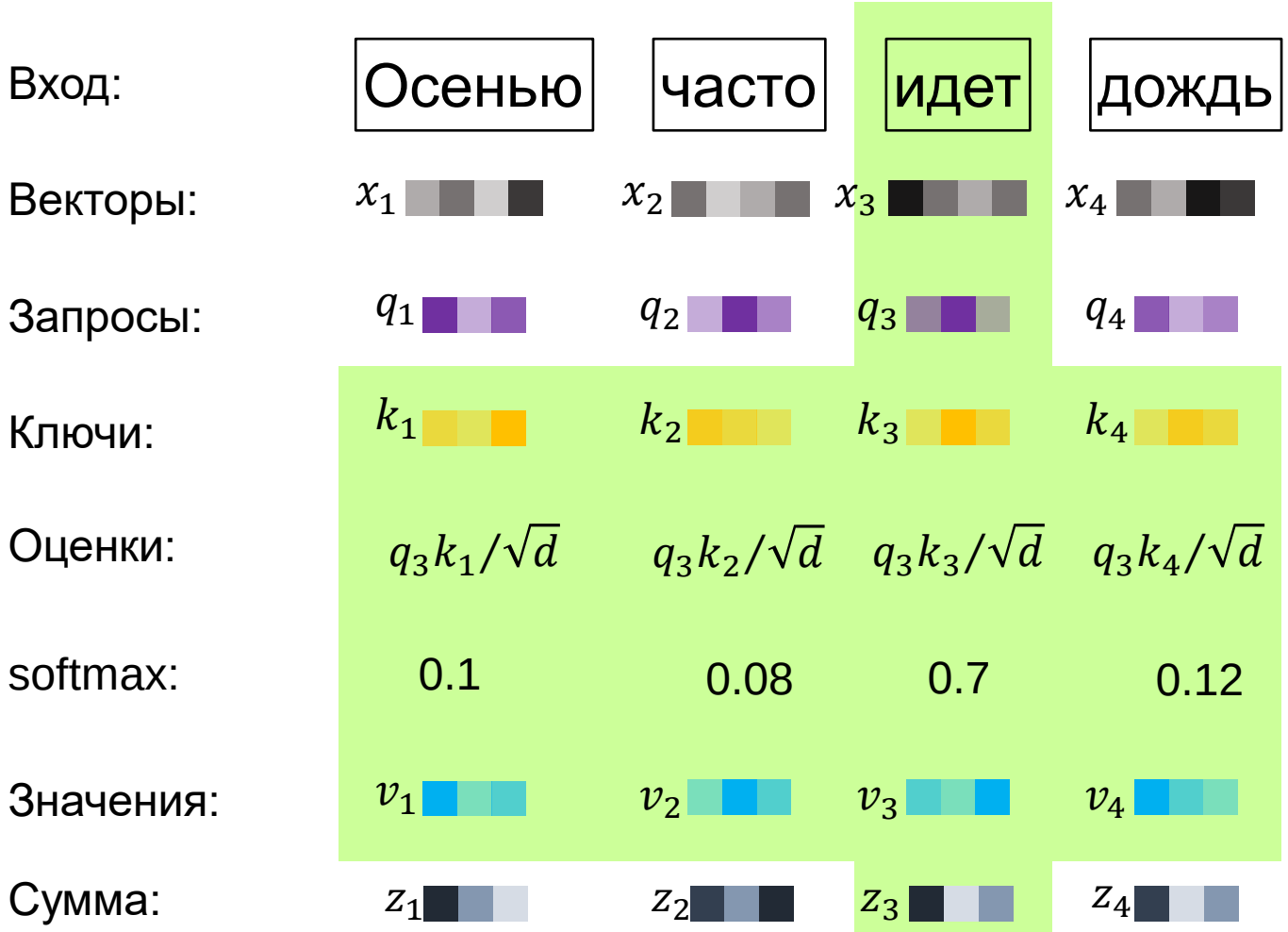
Scaled Dot-Product Self-attention



Scaled Dot-Product Self-attention



Scaled Dot-Product Self-attention



Scaled Dot-Product Self-attention

Вход: Осенью часто идет дождь

Векторы: x_1 x_2 x_3 x_4

$$Q = X * W^Q$$

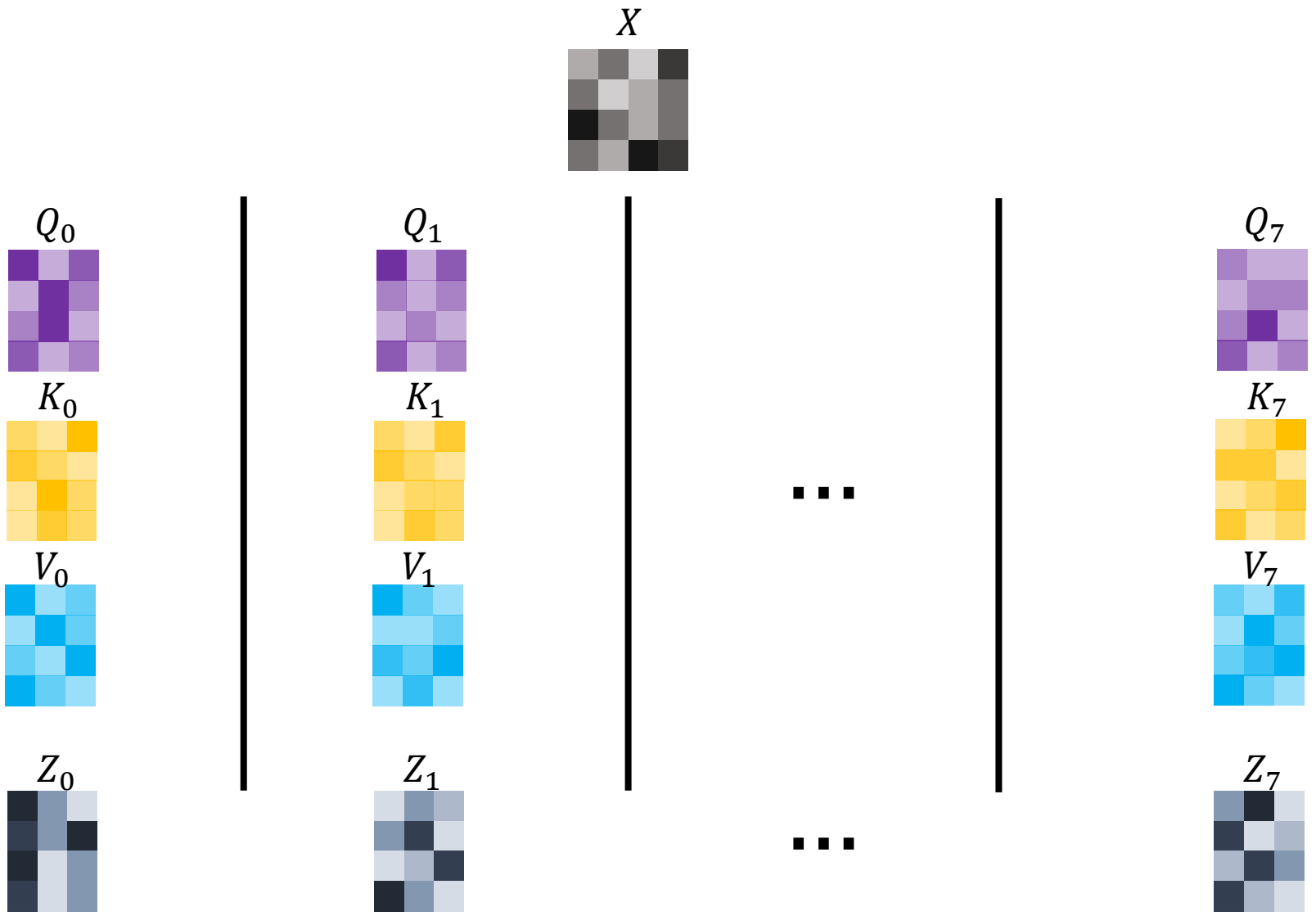
$$K = X * W^K$$

$$V = X * W^V$$

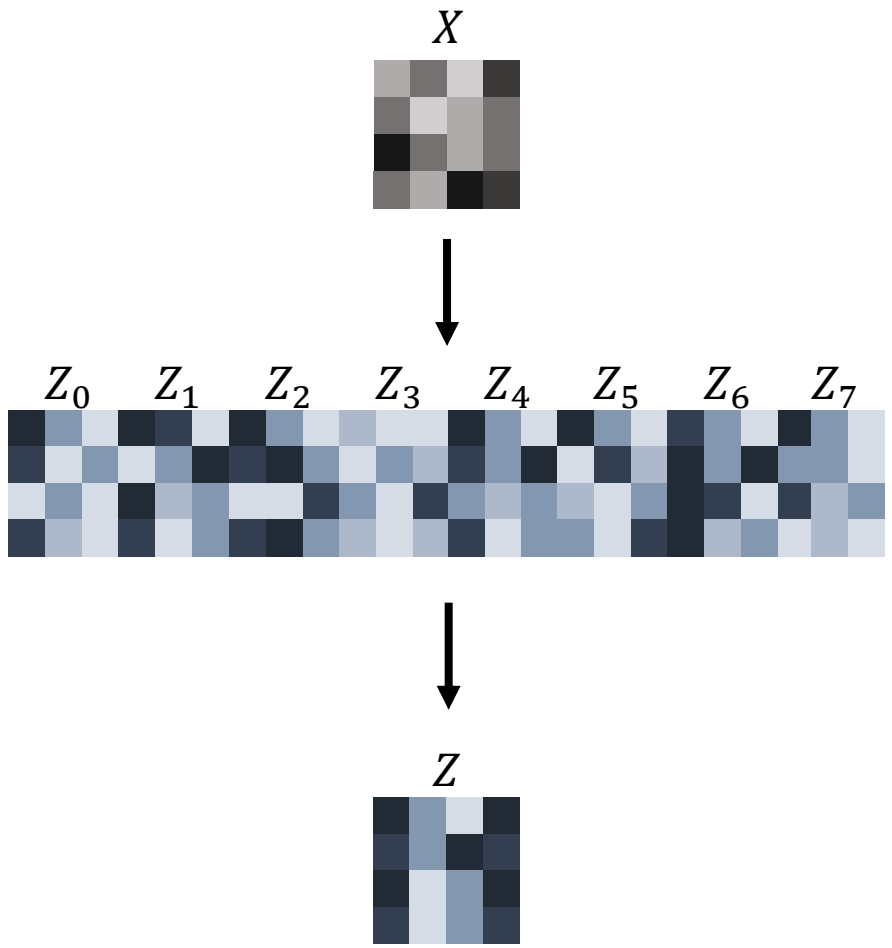
$$Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V$$

z_1 z_2 z_3 z_4

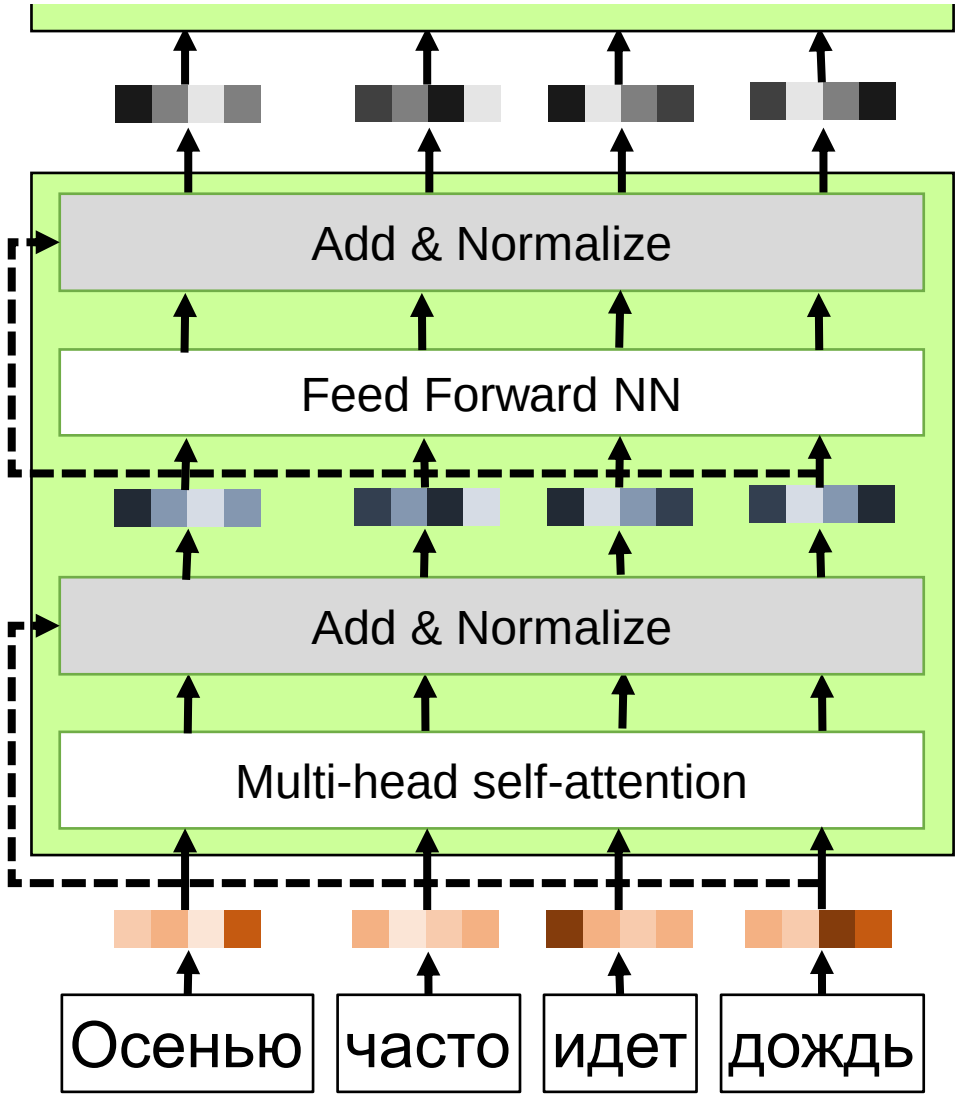
Multi-head attention



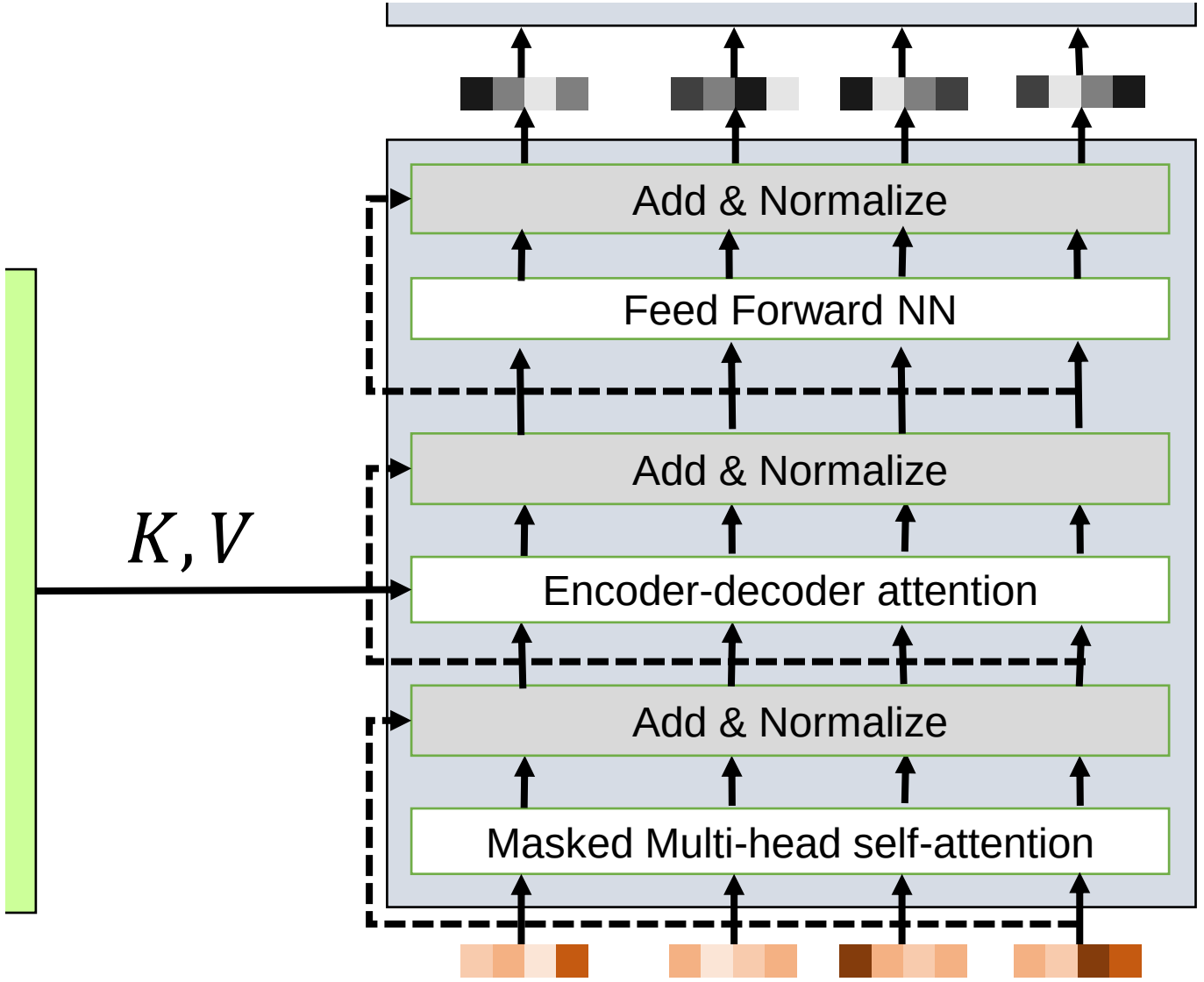
Multi-head attention



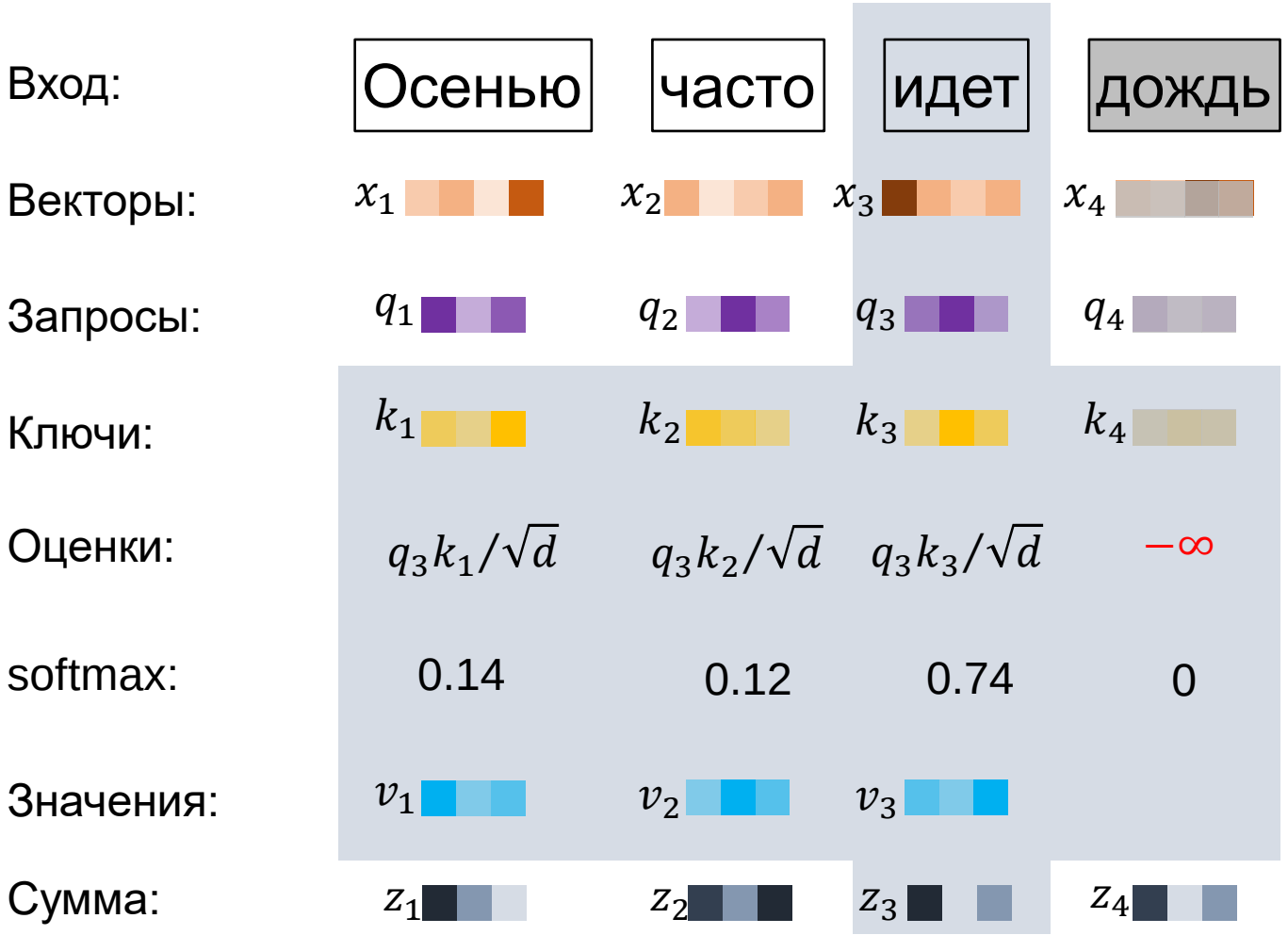
Transformer Encoder



Transformer Decoder

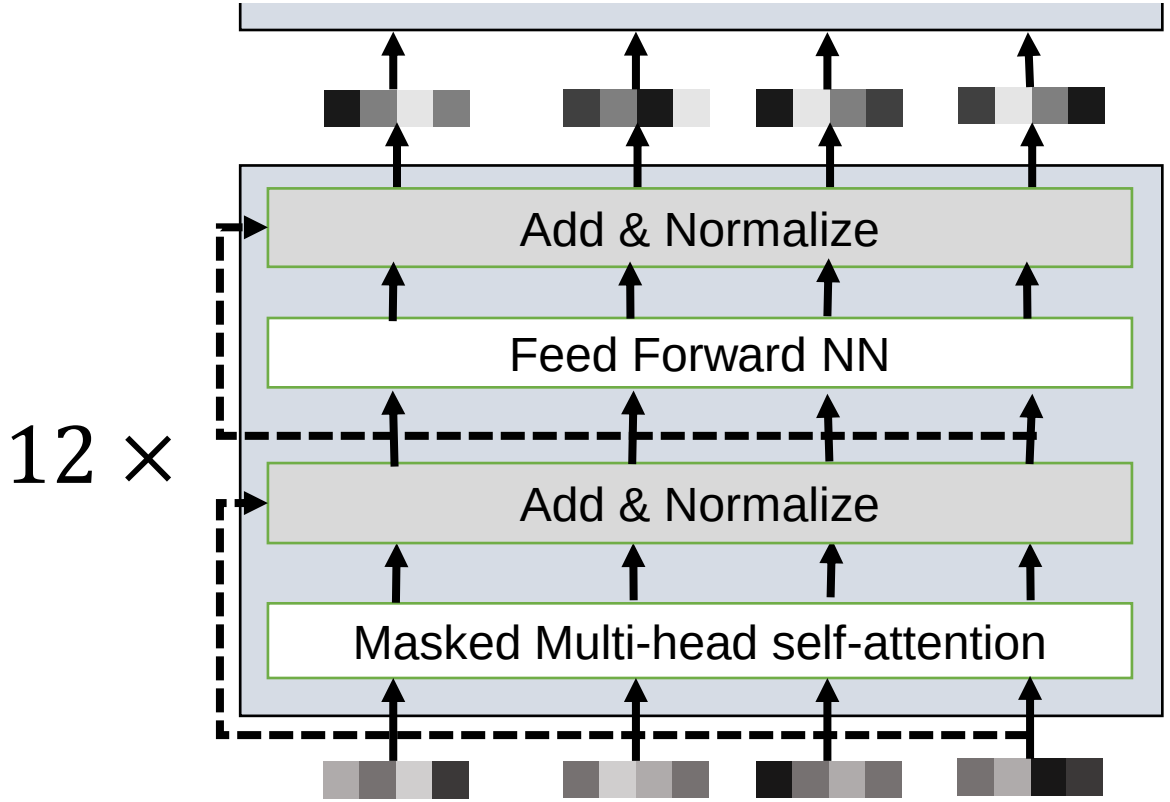


Masked Scaled Dot-Product Self-attention



Языковая модель, основанная на декодере трансформера

- Encoder-decoder attention выкидывается



Radford A. et al. Improving language understanding by generative pre-training. – 2018

GPT (Generative Pre-Training)

ИСП РАН

- Предобучение на корпусе $\mathcal{U}$ (без учителя)

$$h_0 = UW_e + W_p;$$
$$h_i = \text{transformer}(h_{i-1}), i = \overline{1,n}$$
$$P(u) = \text{softmax}(h_n W_e^T)$$

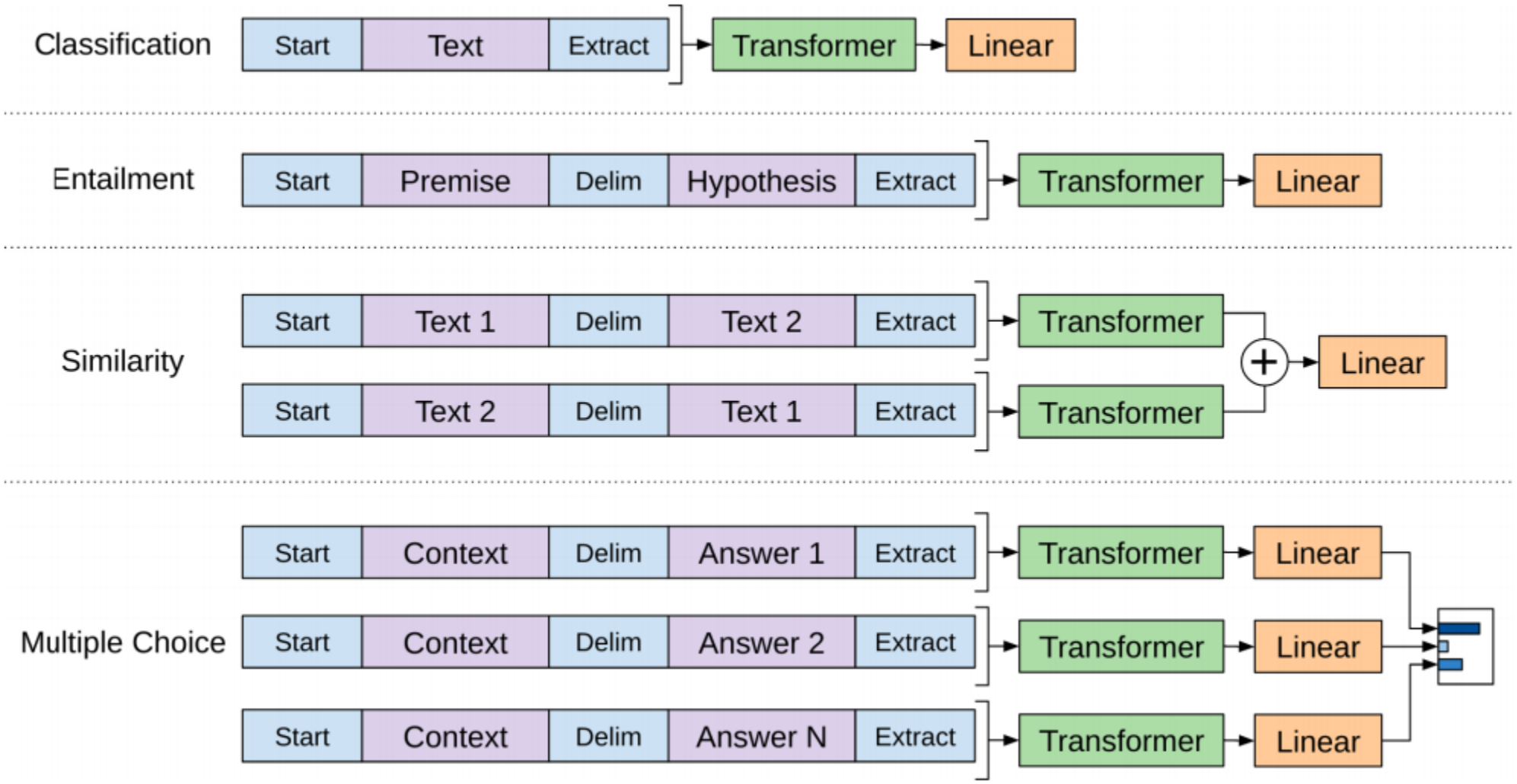
$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

- Донастройка под целевую задачу на корпусе $\mathcal{C}$ (с учителем)

$$P(y|x^1, \dots, x^m) = \text{softmax}\left(h_l^m W_y\right)$$
$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \dots, x^m)$$

$$L(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C});$$

GPT (Generative Pre-Training)



*Картинка из [оригинальной статьи]

Основная идея:

- Большинство задач решается оценкой распределения:

$$p(output|input)$$

- Хочется построить модель, которая будет решать множество задач. То есть оценивать:

$$p(output|input, task)$$

- Задачу можно формулировать словами и подавать языковой модели вместе с $input$

Radford A. et al. Language models are unsupervised multitask learners – 2019

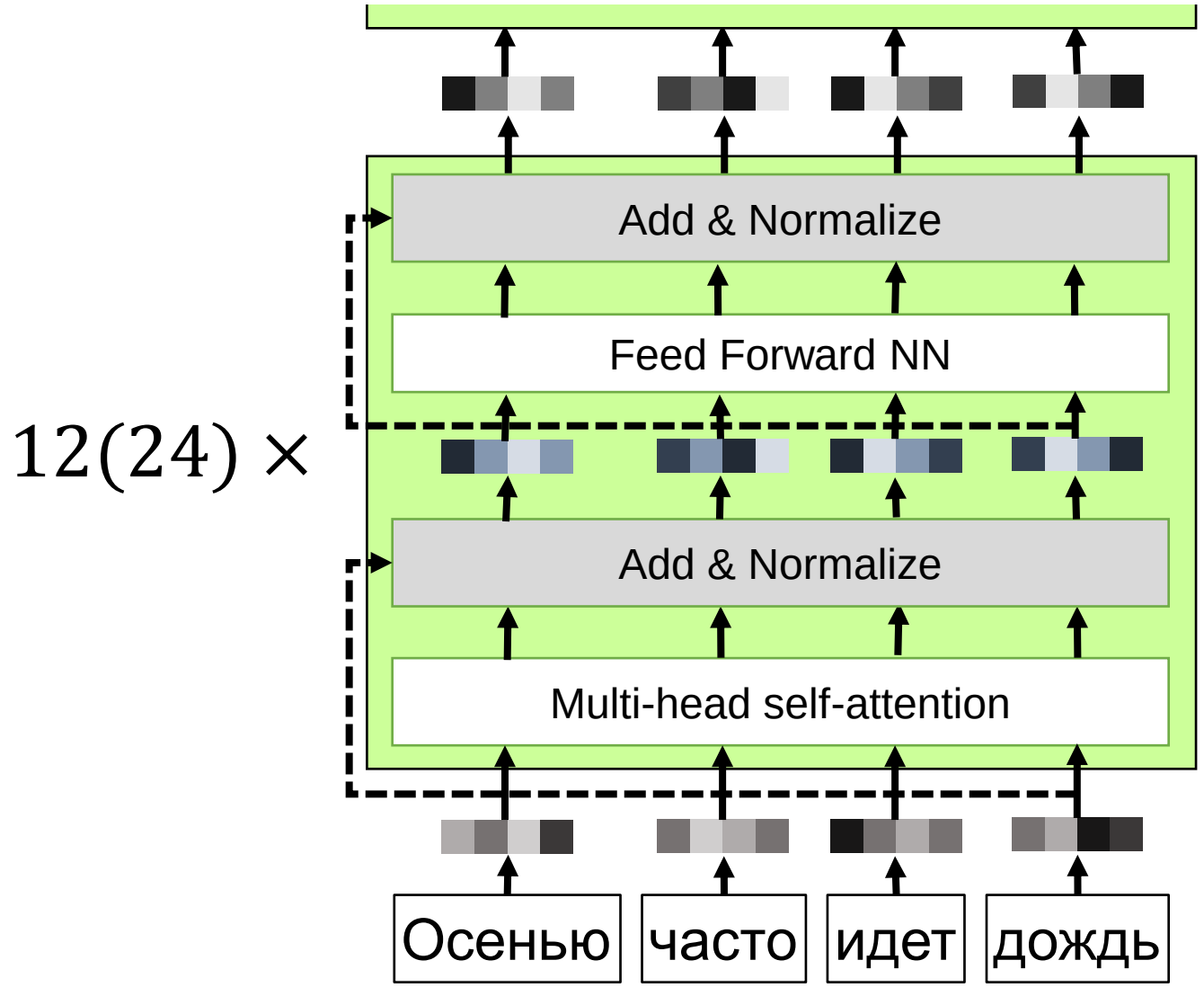
Очень похожа на GPT, но :

- Больше параметров (1.5 миллиарда параметров [48 слоев трансформера, размерность векторов 1600] vs 117 миллионов в GPT)
- Больше корпус для обучения (40 GB текста)
- Другая токенизация: BPE по байтам, а не по символам
- Небольшие изменения архитектуры сети:
  - Передвинут Layer normalization
  - Изменена инициализация

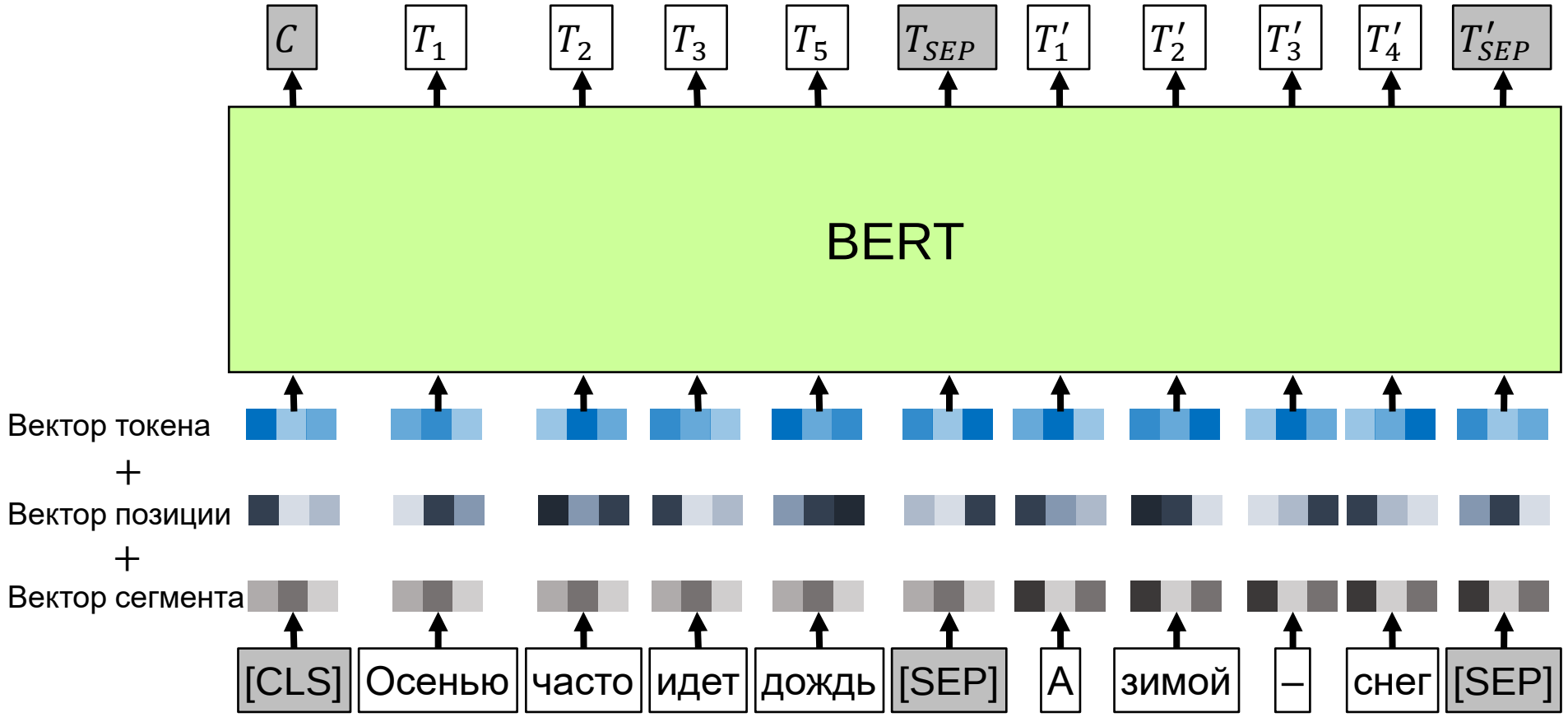
Практически ничем не отличается от GPT-2, но

- Еще больше параметров (175 миллиардов [96 слоев трансформера, размерность векторов – 12288])
- Еще больше корпус для обучения (570 GB текста)
- Еще больше контекст (2048 токенов)

Brown T. B. et al. Language models are few-shot learners – 2020.

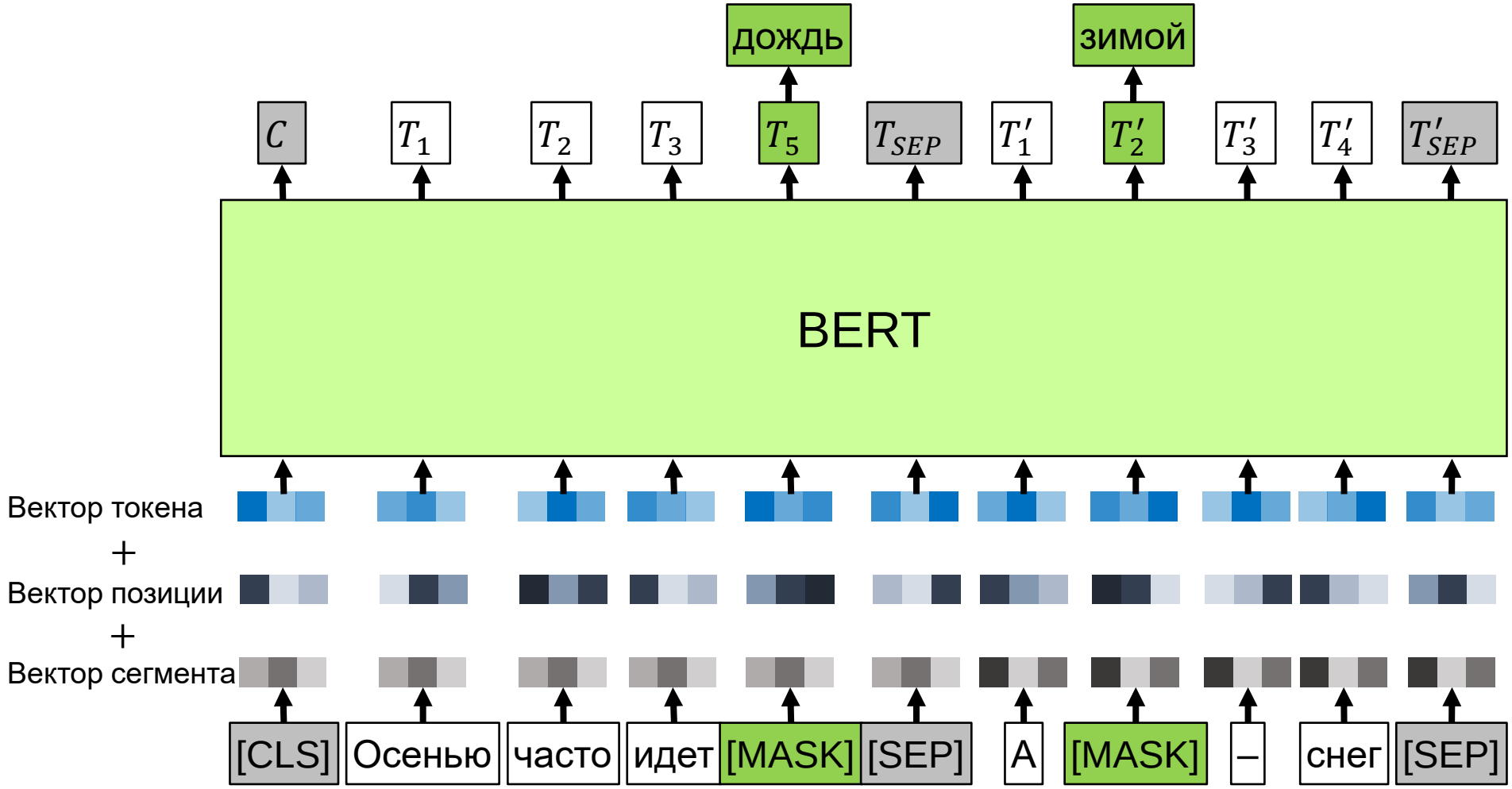


Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding – 2018.


C
T_1
T_2
T_3
T_5
T_{SEP}
T'_1
T'_2
T'_3
T'_4
T'_{SEP}
BERT
Вектор токена
+
Вектор позиции
+
Вектор сегмента
[CLS]
Осенью
часто
идет
дождь
[SEP]
А
зимой
–
снег
[SEP]

BERT pretraining (masked LM)

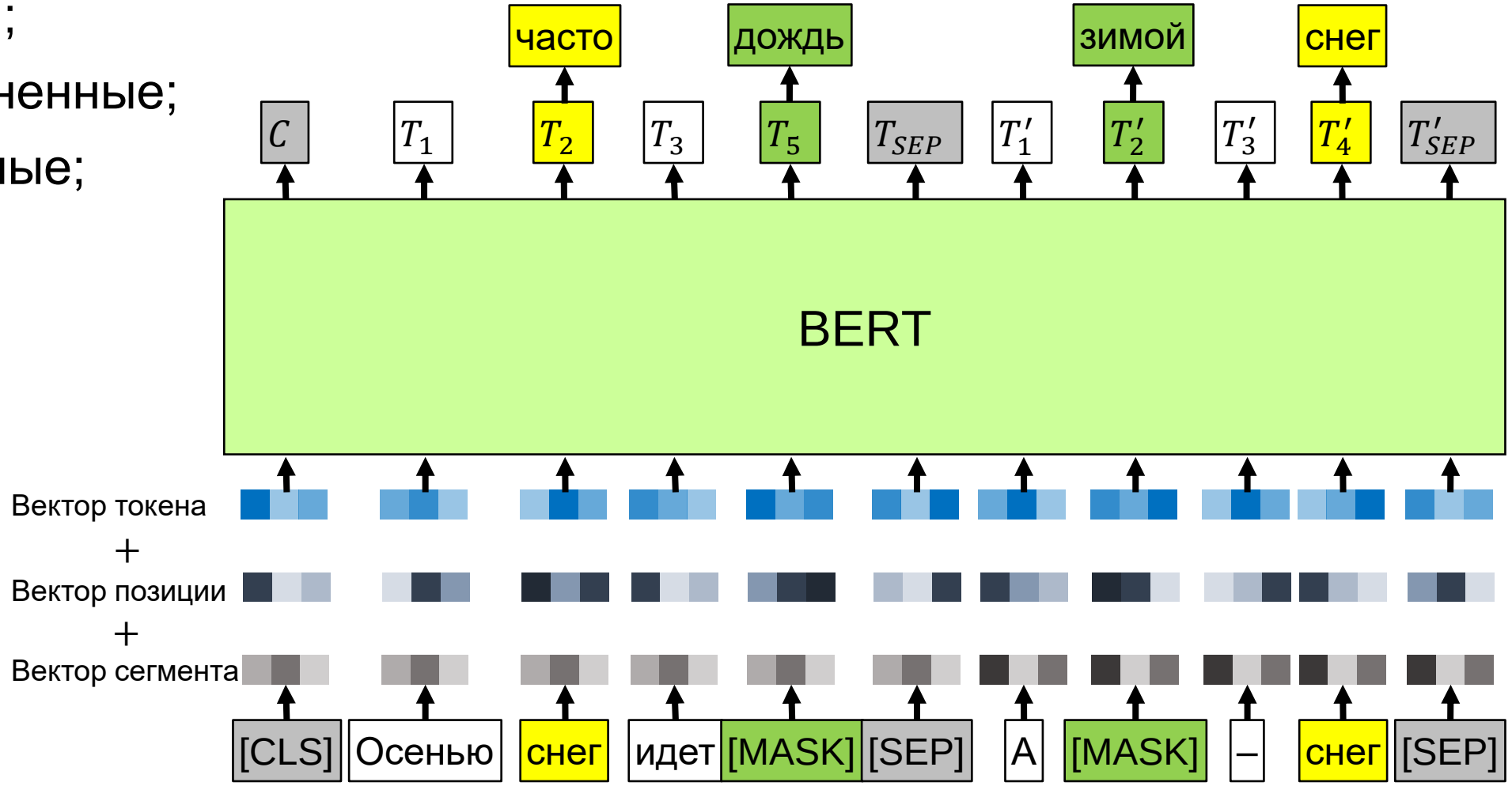
15% токенов маскируются



BERT pretraining (masked LM)

15% токенов маскируются

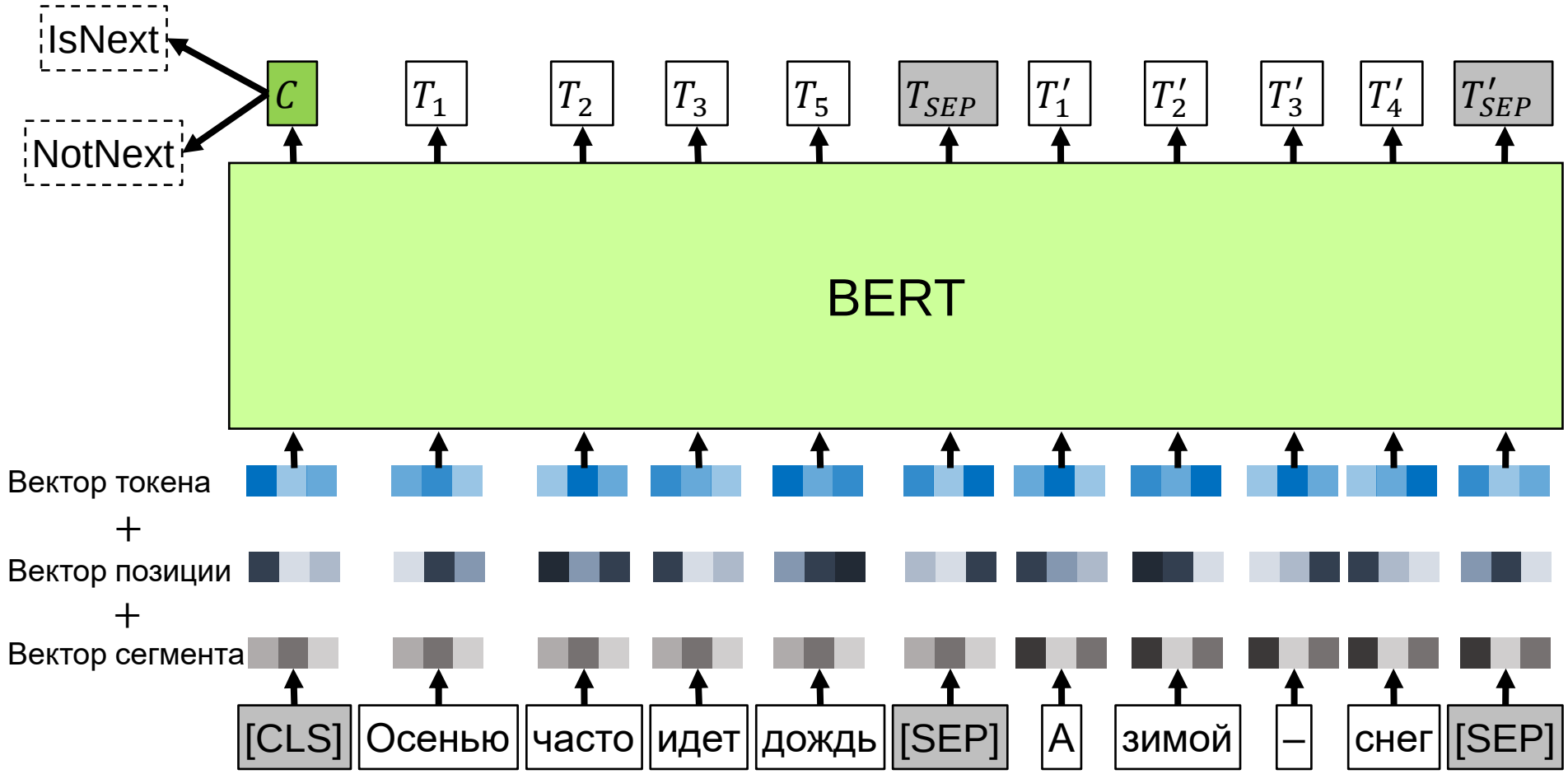
- 80% - [MASK];
- 10% - неизмененные;
- 10% - случайные;



BERT pretraining (NSP)

ИСП РАН

Предсказывается, является ли фрагмент B следующим за фрагментом A



Литература

- Peters M. E. et al. Semi-supervised sequence tagging with bidirectional language models – 2017.
- Peters M. E. et al. Deep contextualized word representations – 2018.
- Vaswani A. et al. Attention is all you need – 2017.
- Radford A. et al. Improving language understanding by generative pre-training – 2018.
- Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding – 2018.
- Radford A. et al. Language models are unsupervised multitask learners – 2019.
- Brown T. B. et al. Language models are few-shot learners – 2020.