

Глубинное обучение

Лекция 15
Трансформеры

Михаил Гущин
mhushchyn@hse.ru

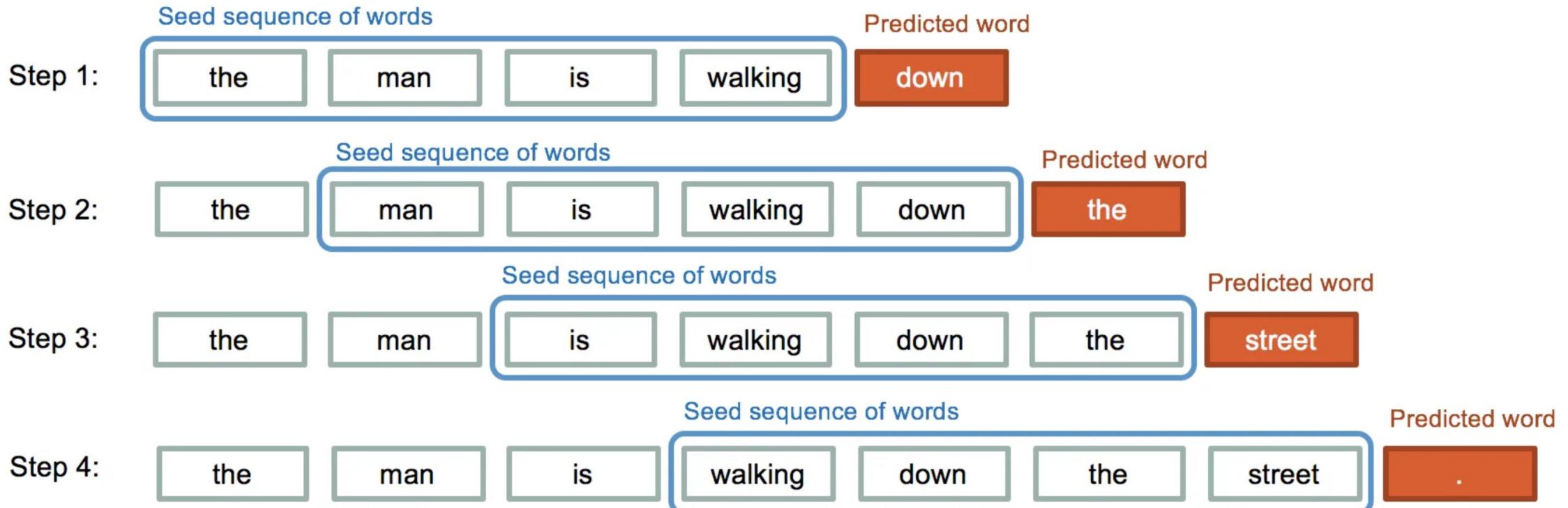
НИУ ВШЭ, 2024



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Механизмы внимания

Генерация текста



Link: <https://bansalh944.medium.com/text-generation-using-lstm-b6ced8629b03>

Attention

- ▶ В общем виде механизм внимания представляем в виде слоя нейронной сети:

$$\mathbf{Z} = \text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^T}{\sqrt{d}}\right) \mathbf{V}$$

- ▶ $Q \in R^{(n \times d)}$ - последовательность n **входных** запросов (queries) размера d .
Например, это входная последовательность токенов (слов).
- ▶ $K \in R^{(m \times d)}$ - последовательность m ключей (keys) размера d . Ключи нам даны, либо мы их предварительно как-то вычисляем.
- ▶ $V \in R^{(m \times v)}$ - последовательность m значений (values) размера v . Значения нам даны, либо мы их предварительно как-то вычисляем.
- ▶ $Z \in R^{(n \times v)}$ - последовательность n **выходных** значений размера v .

Self-Attention

- ▶ Пусть дана последовательность n токенов $X \in R^{(n \times k)}$. Тогда

$$Q = XW_Q, \quad W_Q \in R^{(k \times d)}$$

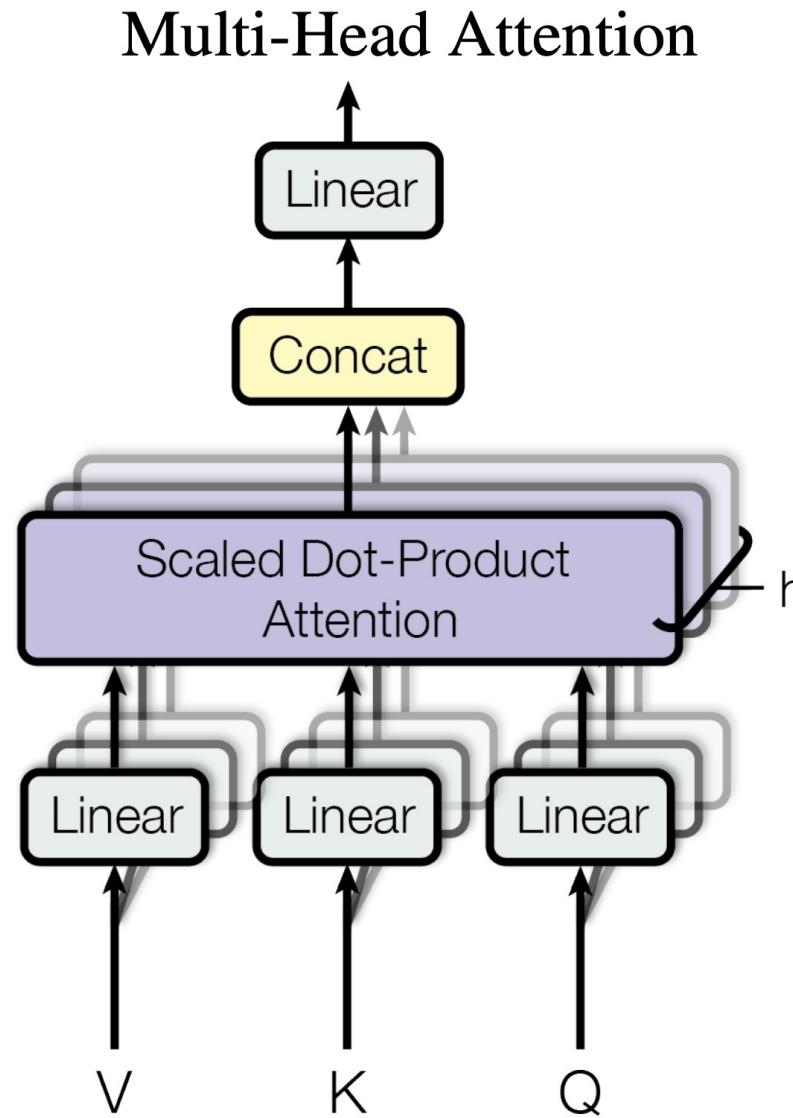
$$K = XW_K, \quad W_K \in R^{(k \times d)}$$

$$V = XW_V, \quad W_V \in R^{(k \times v)}$$

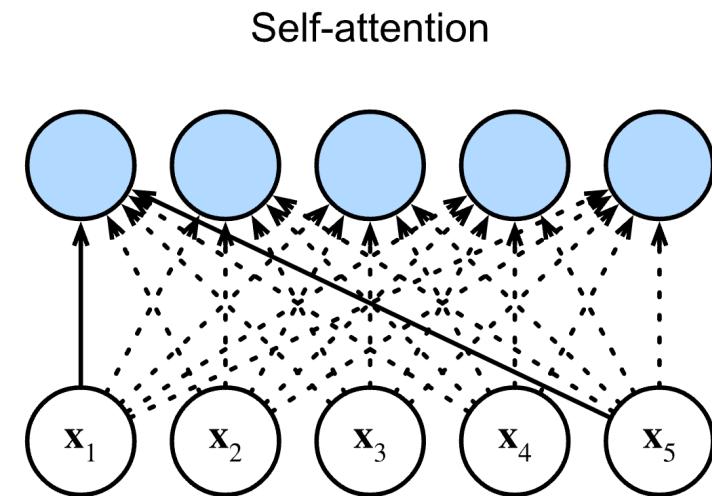
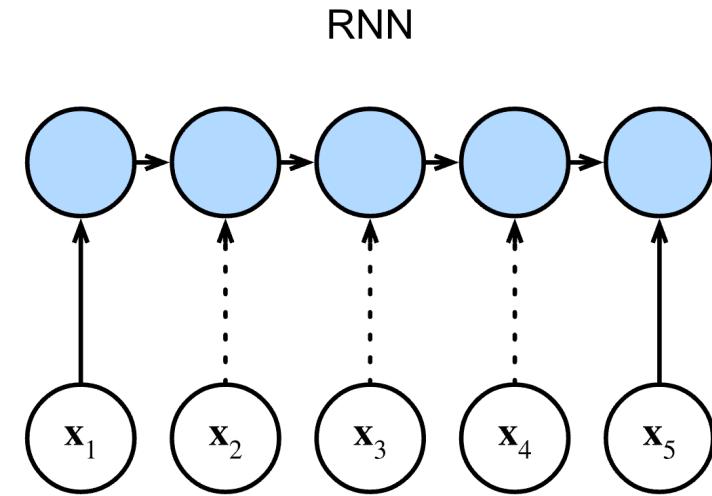
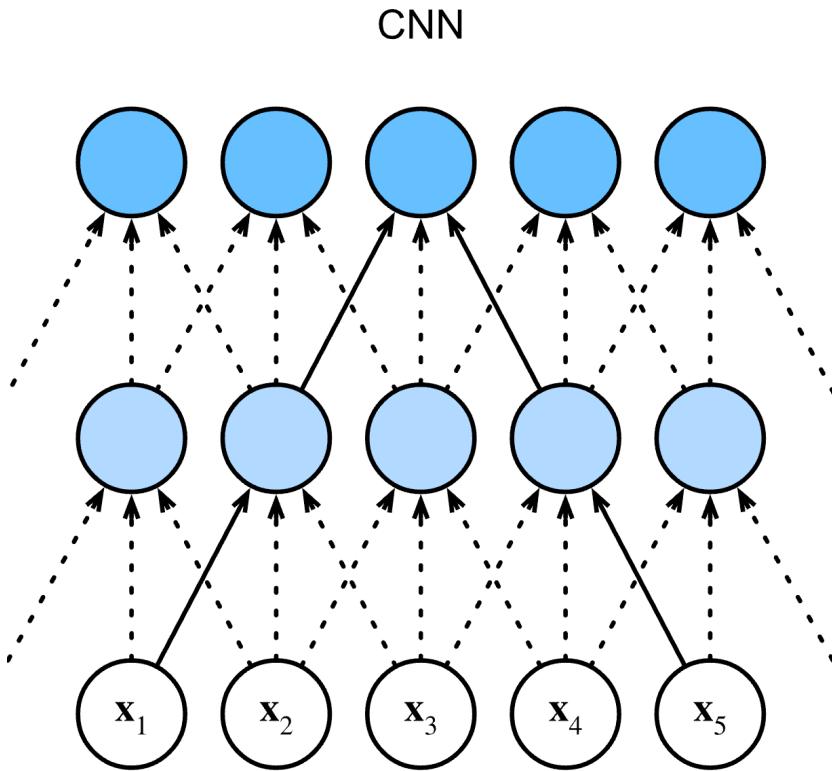
$$\textcolor{red}{Z} = \text{softmax}\left(\frac{\textcolor{blue}{Q}K^T}{\sqrt{d}}\right)V$$

- ▶ W_Q, W_K, W_V - обучаемые веса, полно связные нейронные слои
- ▶ $Q \in R^{(n \times d)}, K \in R^{(n \times d)}, V \in R^{(n \times v)}, Z \in R^{(n \times v)}$

Multi-Head Attention



Self-Attention



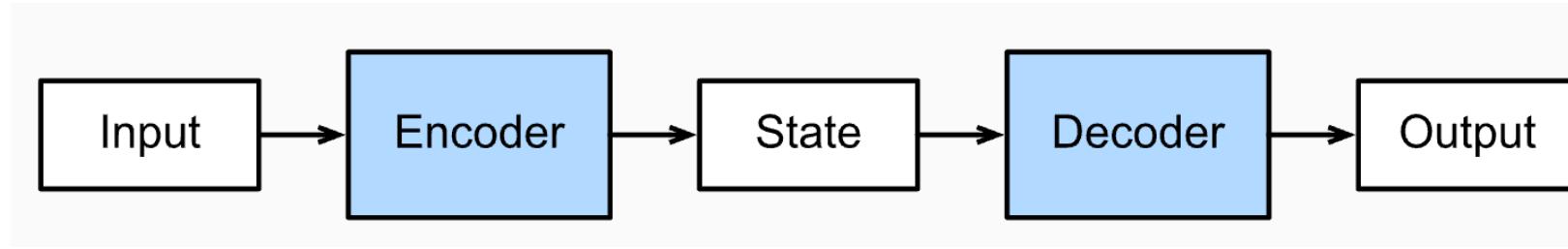
Вопрос

- ▶ Чем self-attention отличается от полносвязного слоя?
- ▶ Чем self-attention отличается от рекуррентного слоя?



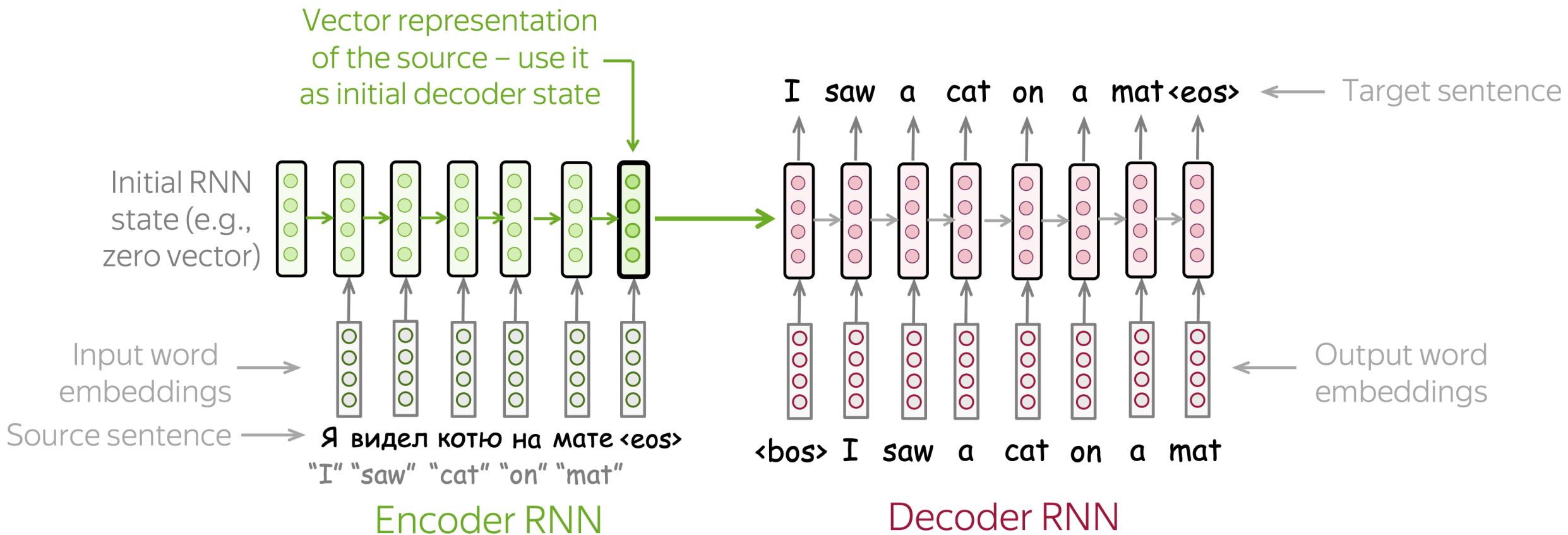
Машинный перевод

Encoder-Decoder



- ▶ На вход подаем текст на одном языке
- ▶ Encoder получает векторное представление «смысла» текста
- ▶ Полученный эмбеддинг подаем на вход Decoder
- ▶ Decoder генерирует перевод с учетом «смысла»

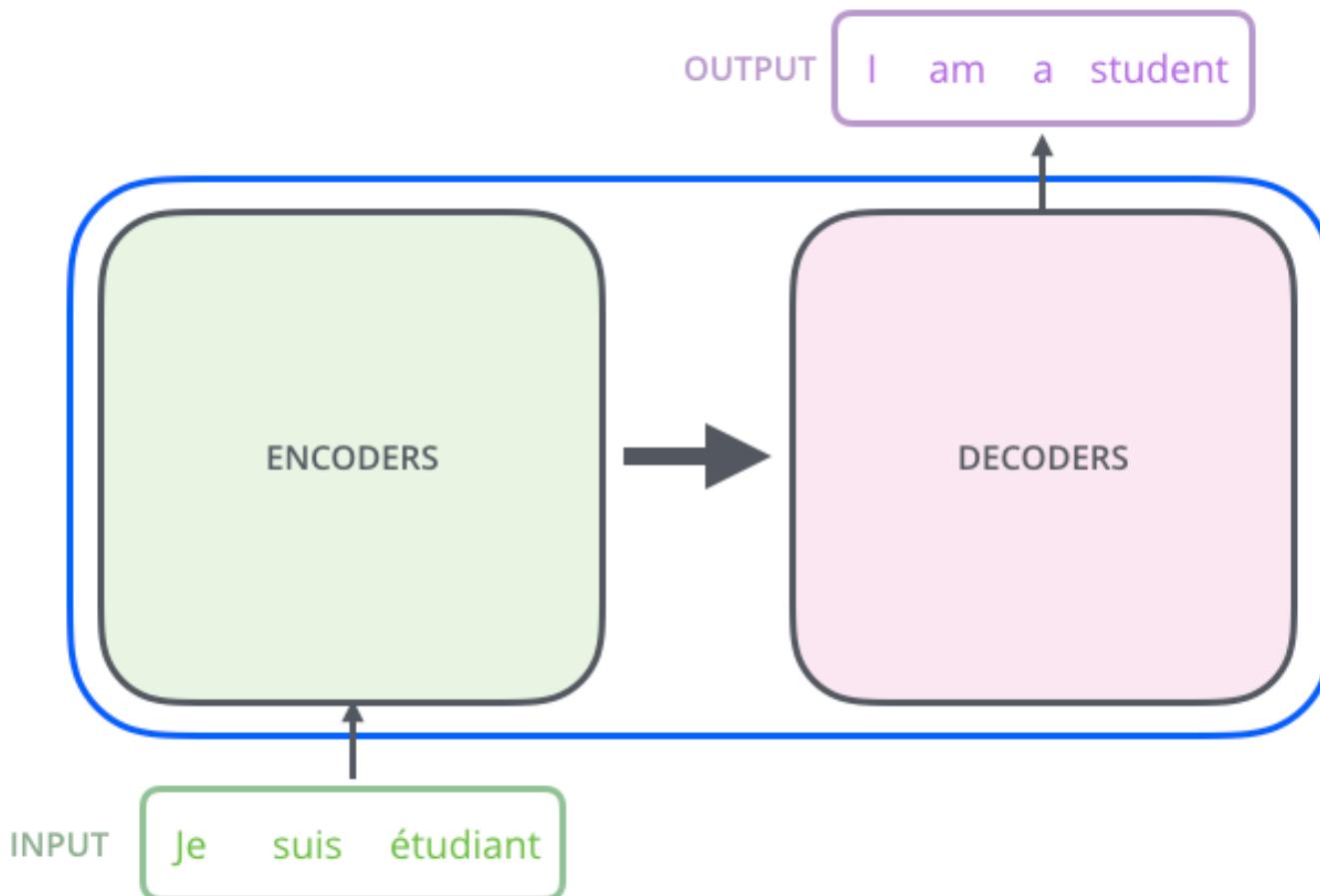
Архитектура RNN для перевода



Источник: https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq

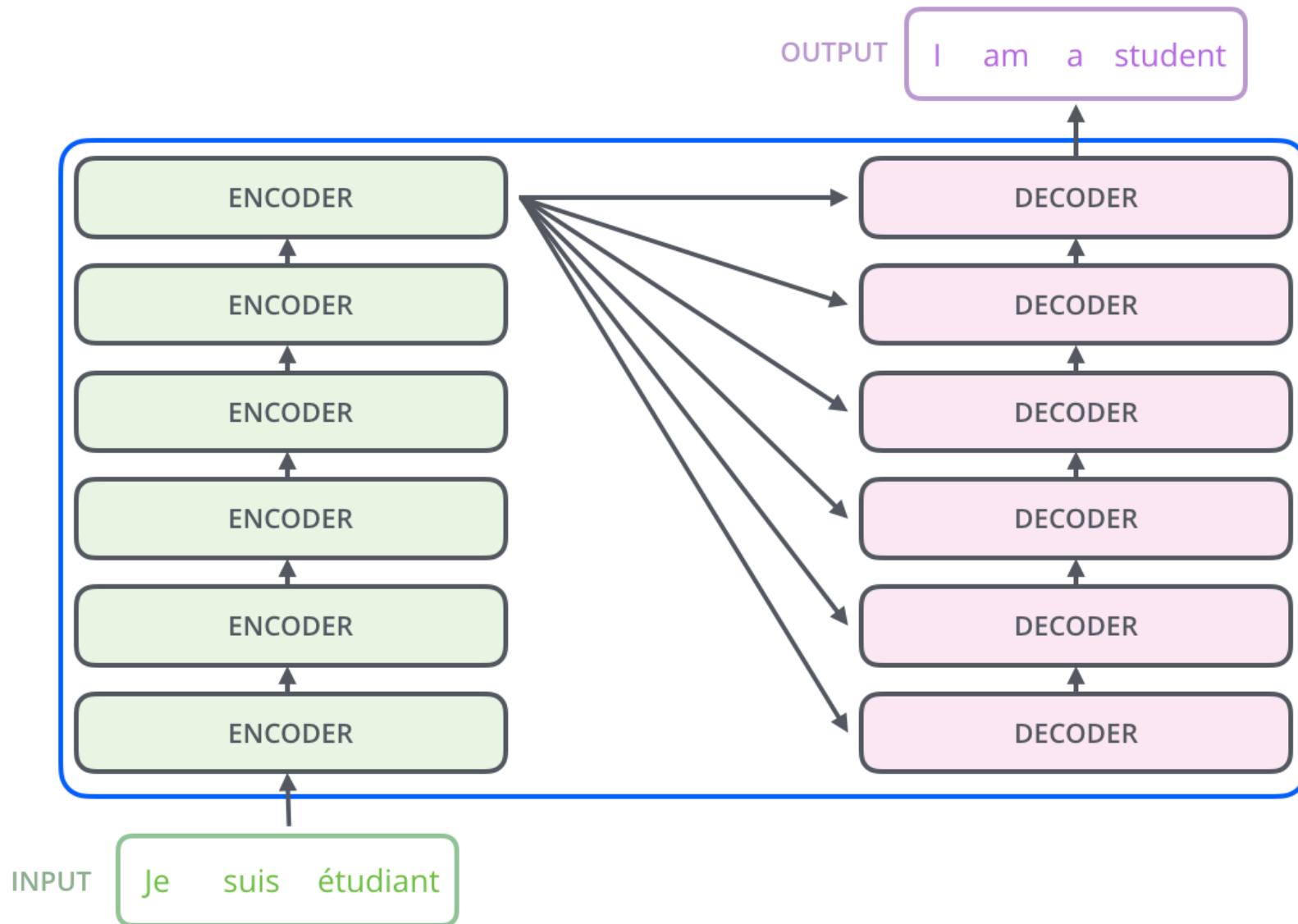
Трансформеры

Общая схема трансформера

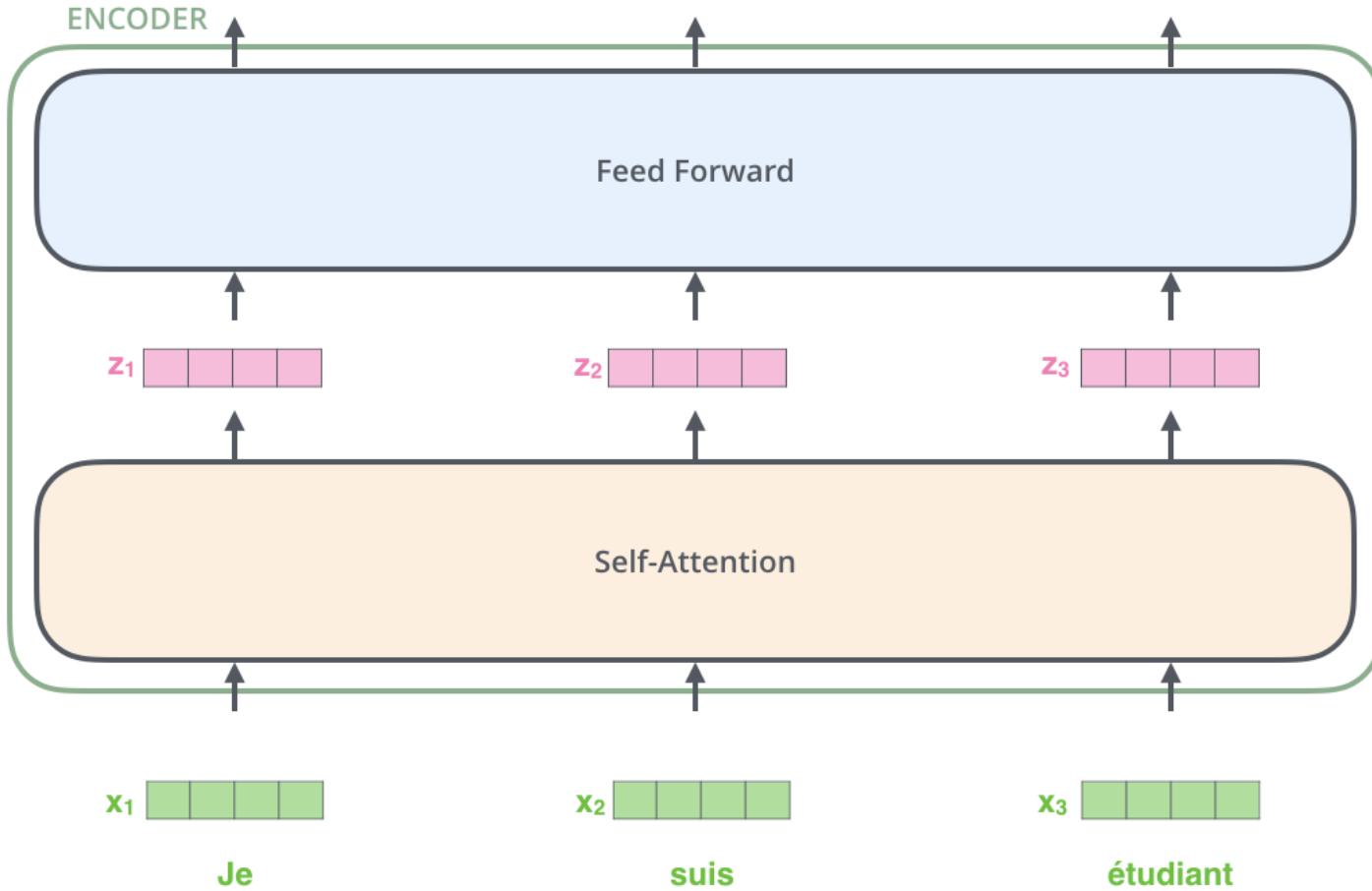


Источник: <https://habr.com/ru/articles/486358/>

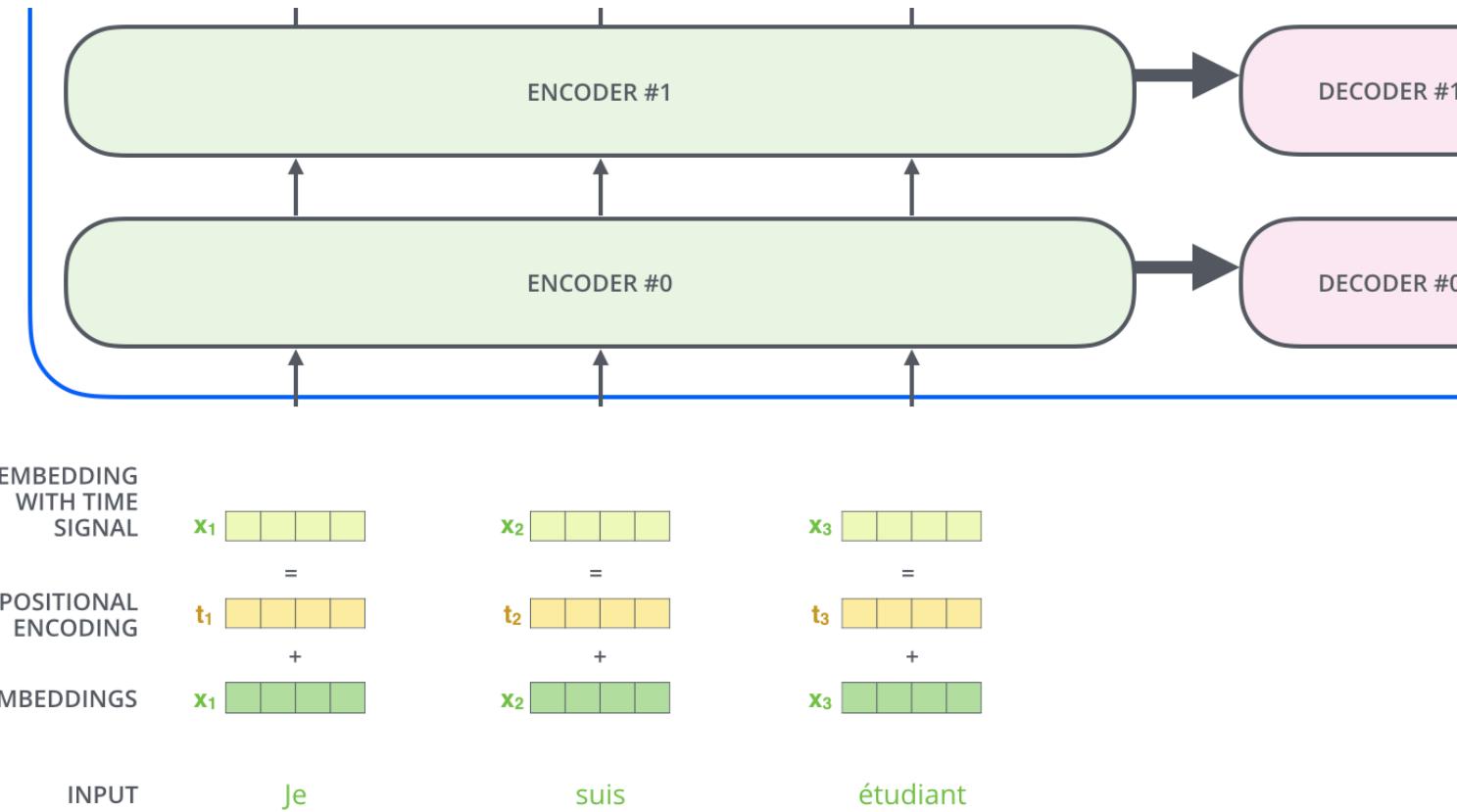
Чуть подробнее



Блок кодировщика в трансформере



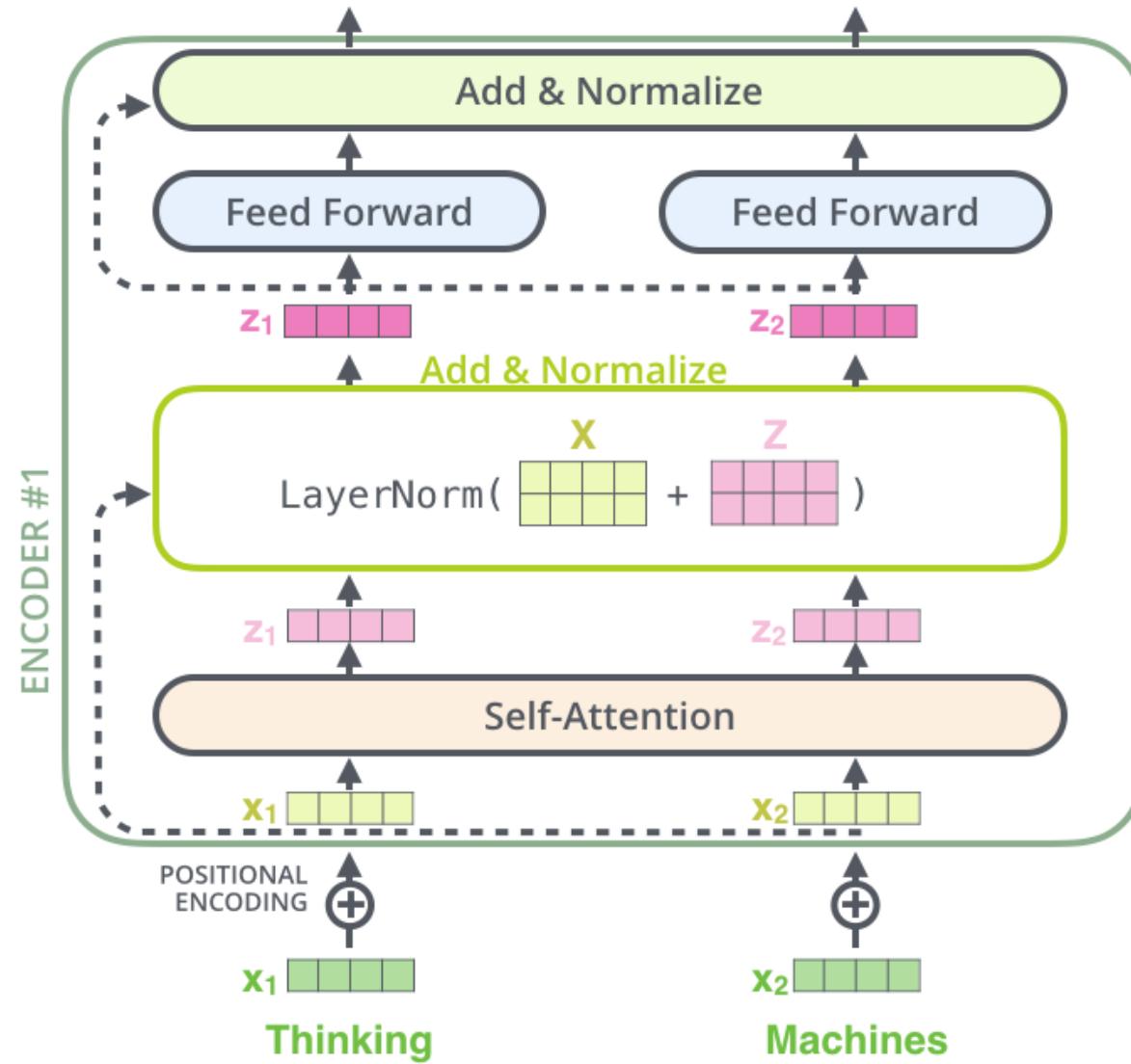
Positional encoding



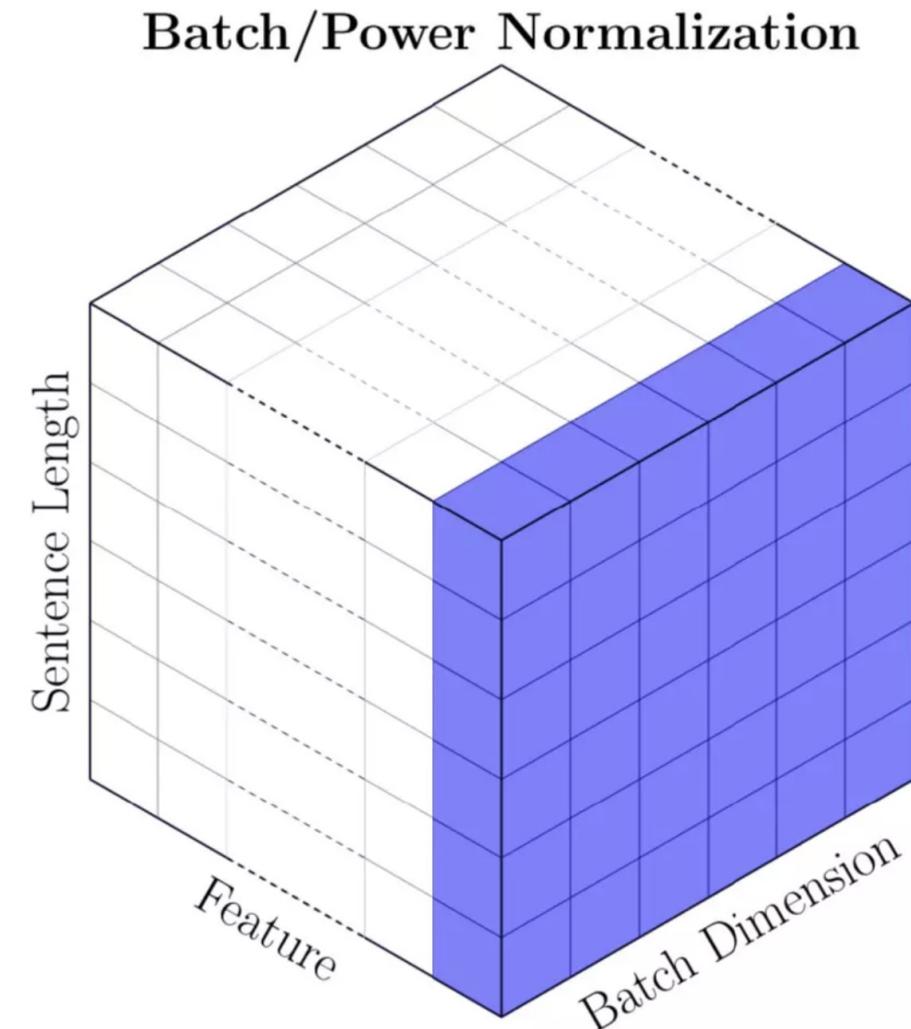
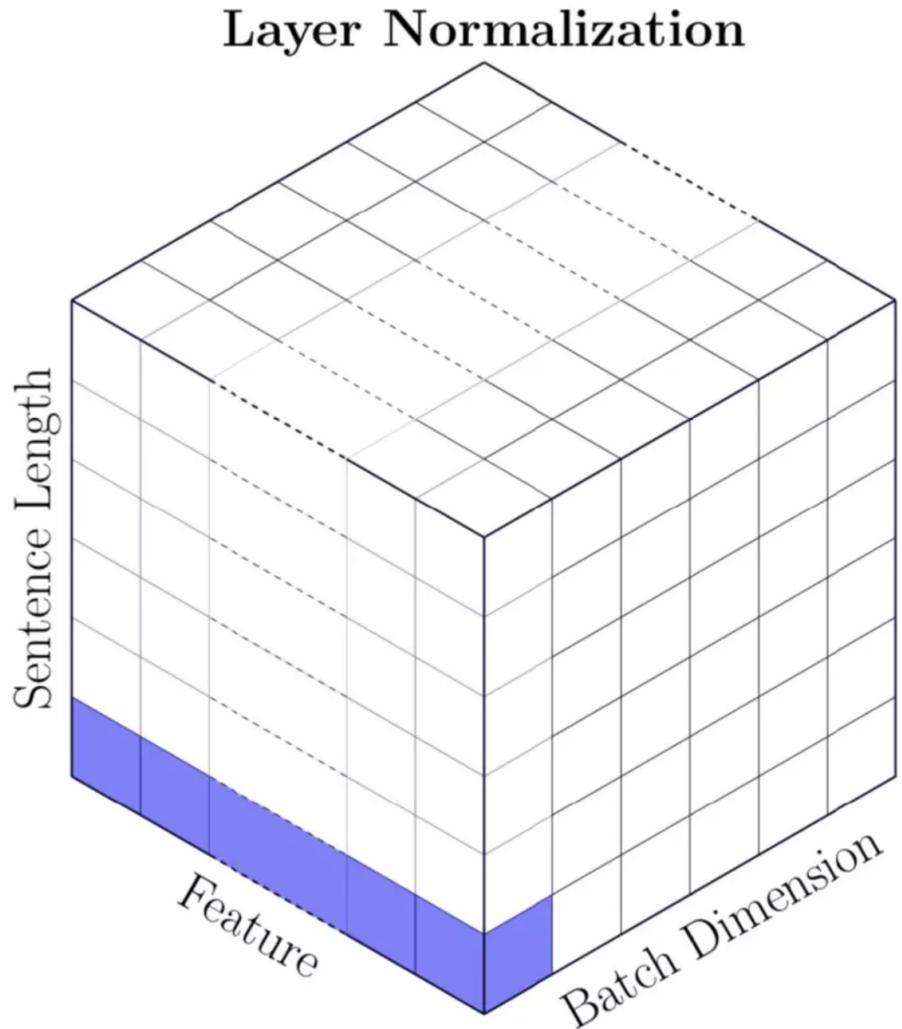
Self-attention не учитывает порядок слов. Чтобы это сделать, добавим **positional encoding**, значение которого зависит от положения токена в последовательности.

Итоговая схема блока кодировщика

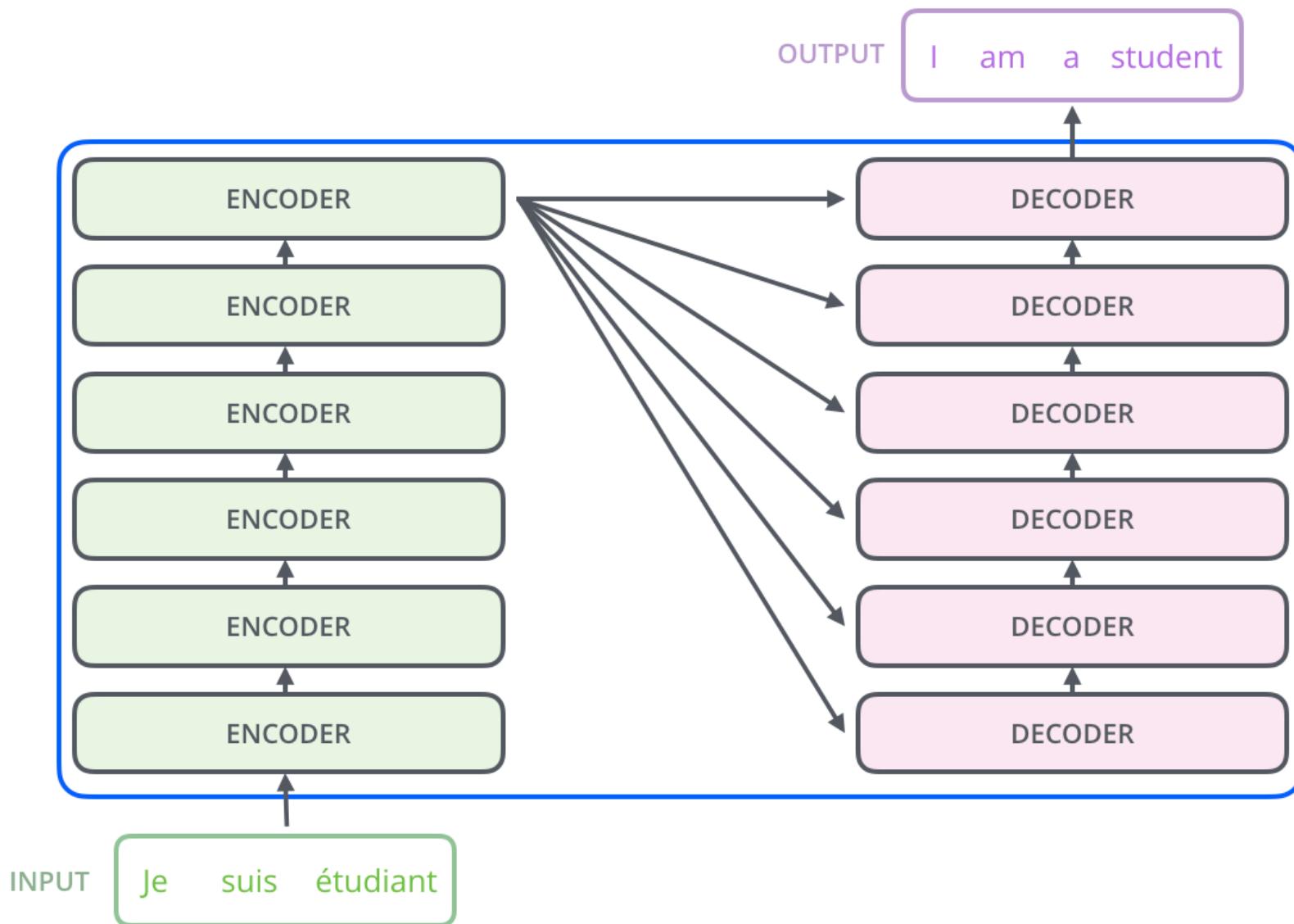
Добавили residual connection и нормализацию по слову



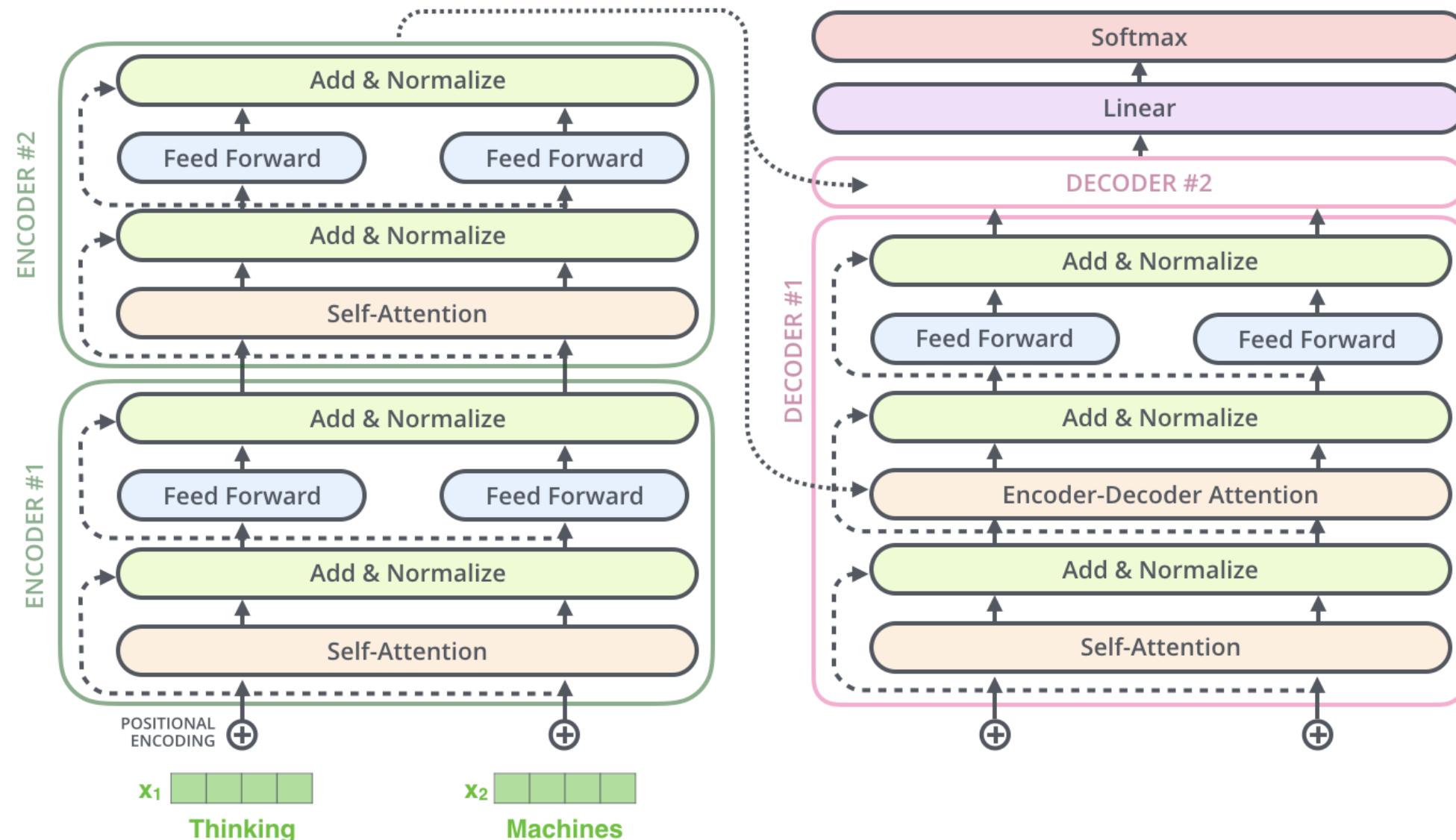
Layer Normalization



Теперь про декодировщики



Связь кодировщик – декодировщик



Encoder-Decoder Attention

- ▶ Выход encoder-decoder attentions считаем по обычной формуле:

$$\mathbf{Z} = \text{softmax} \left(\frac{\mathbf{Q} \mathbf{K}_{enc}^T}{\sqrt{d}} \right) \mathbf{V}_{enc}$$

- ▶ $Q \in R^{(n \times d)}$ - последовательность n **входных** запросов (queries) размера d .
- ▶ $K_{enc} \in R^{(m \times d)}$ - берем из последнего кодировщика (encoder)
- ▶ $V_{enc} \in R^{(m \times v)}$ - берем из последнего кодировщика (encoder)
- ▶ $Z \in R^{(n \times v)}$ - последовательность n **выходных** значений размера v .

Self-attention в декодировщике

- ▶ Разрешается использовать только предыдущие слова

Выходной блок

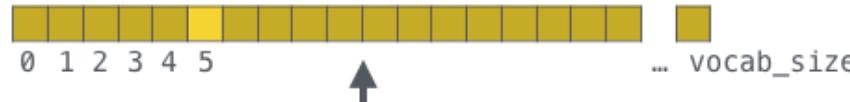
Which word in our vocabulary
is associated with this index?

am

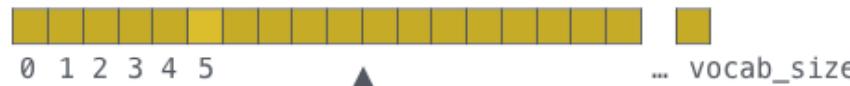
Get the index of the cell
with the highest value
(argmax)

5

log_probs



logits



Decoder stack output



Linear

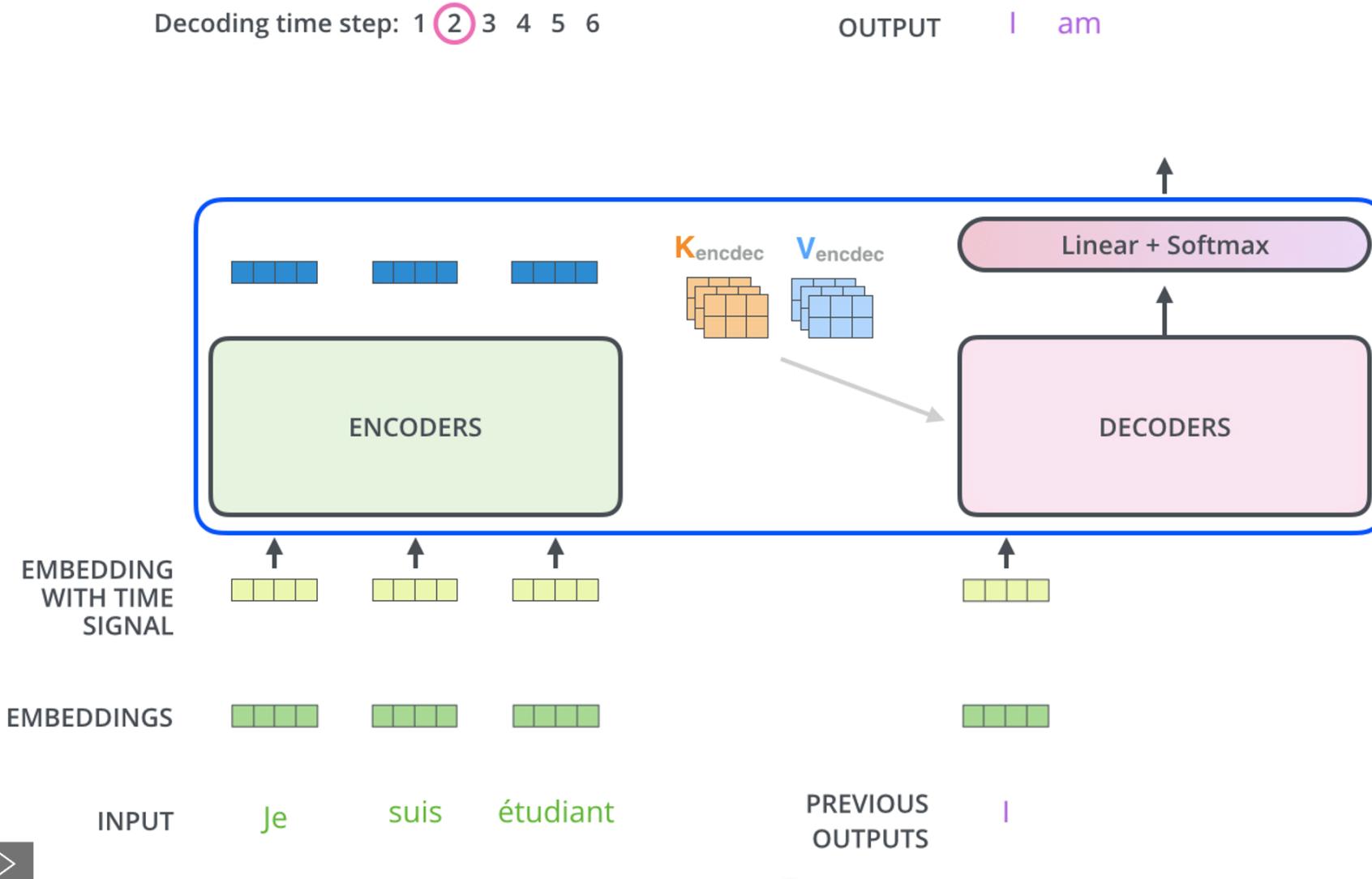


Softmax

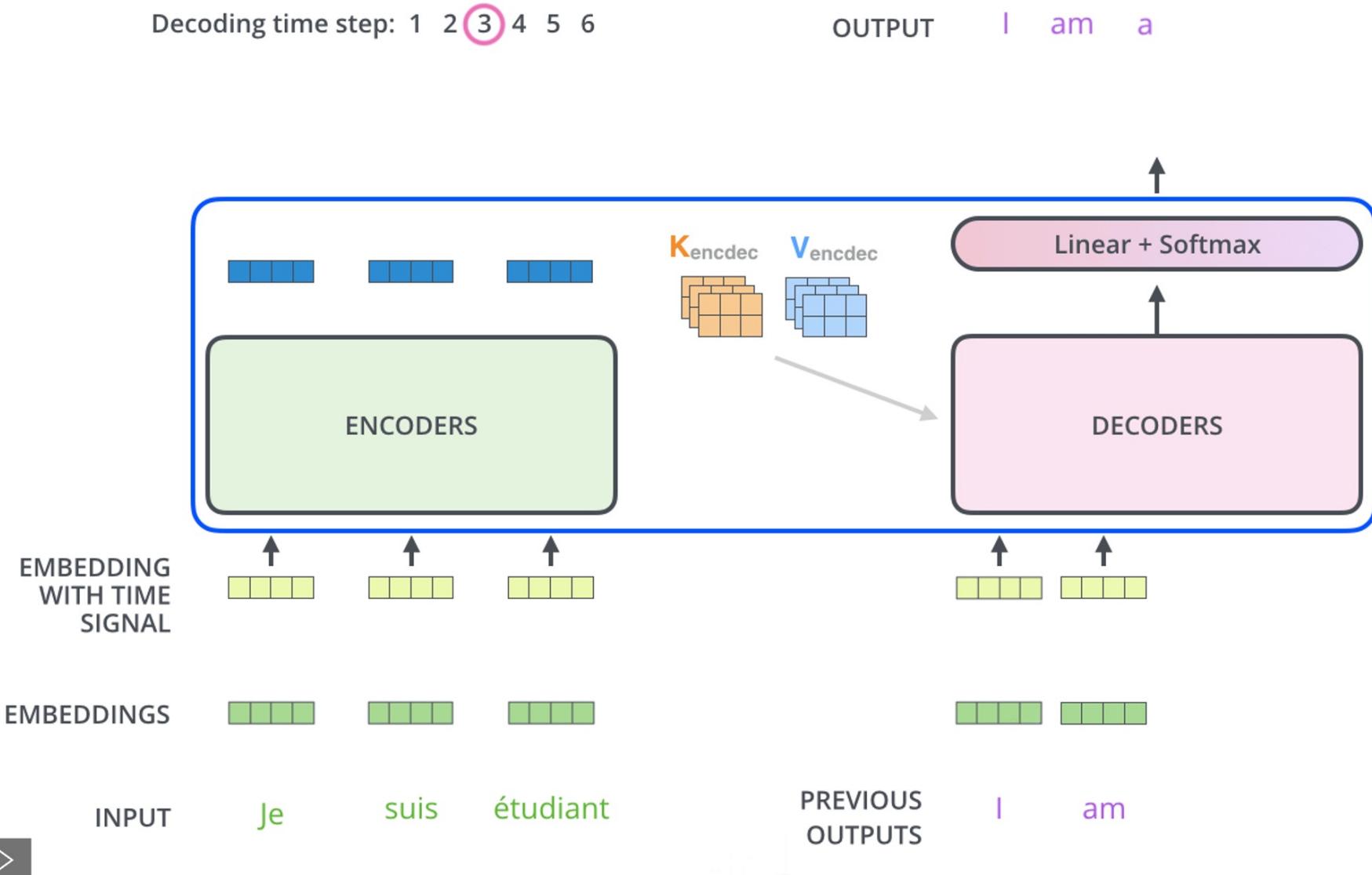
Генерация перевода

- ▶ В случае с машинным переводом — «авторегрессионное» применение:
 - Сначала декодировщик выдаёт одно слово
 - Затем два (первое подаётся как вход)
 - Затем три (первые два подаются ему как вход)
 - И т.д.

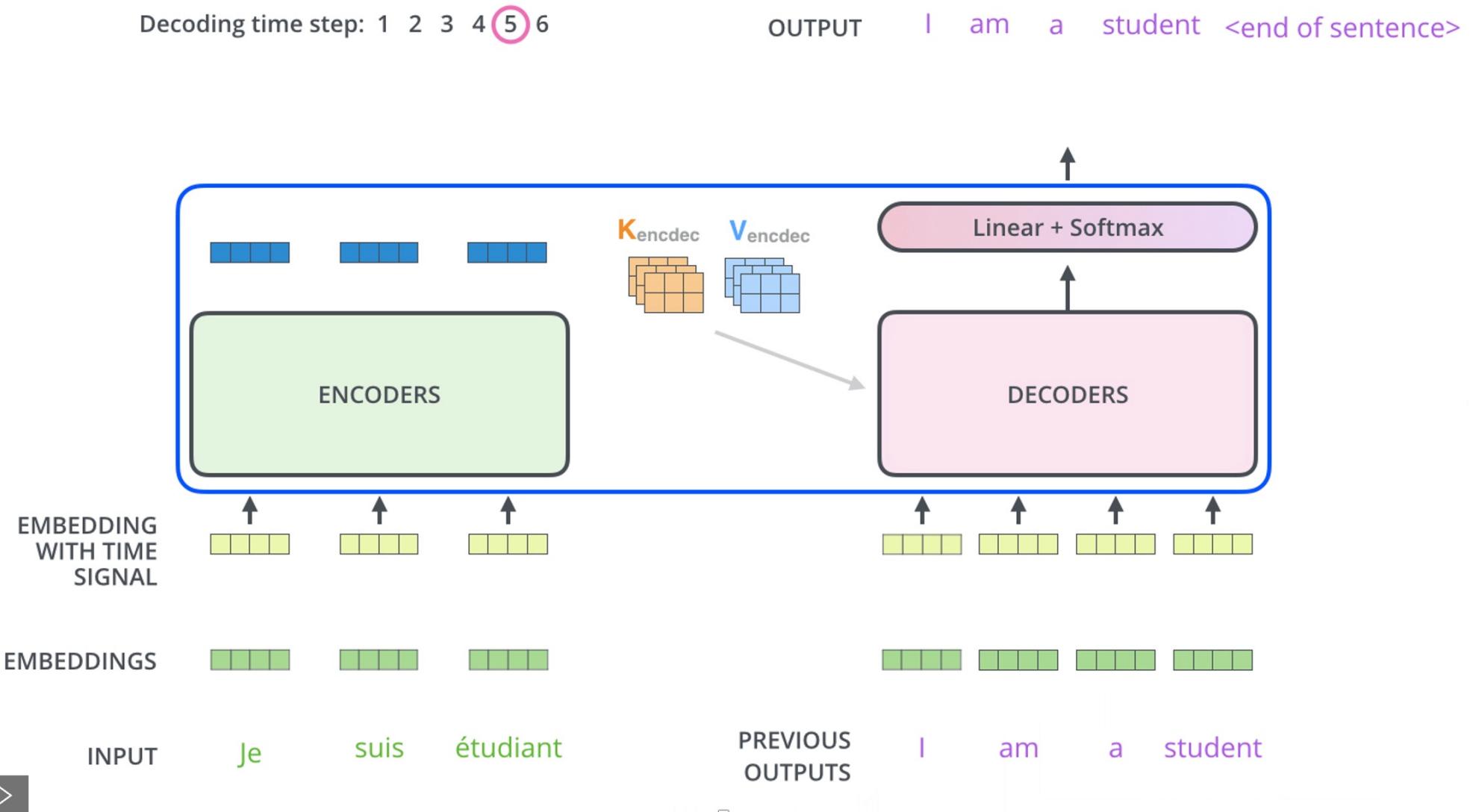
Пример



Пример



Пример

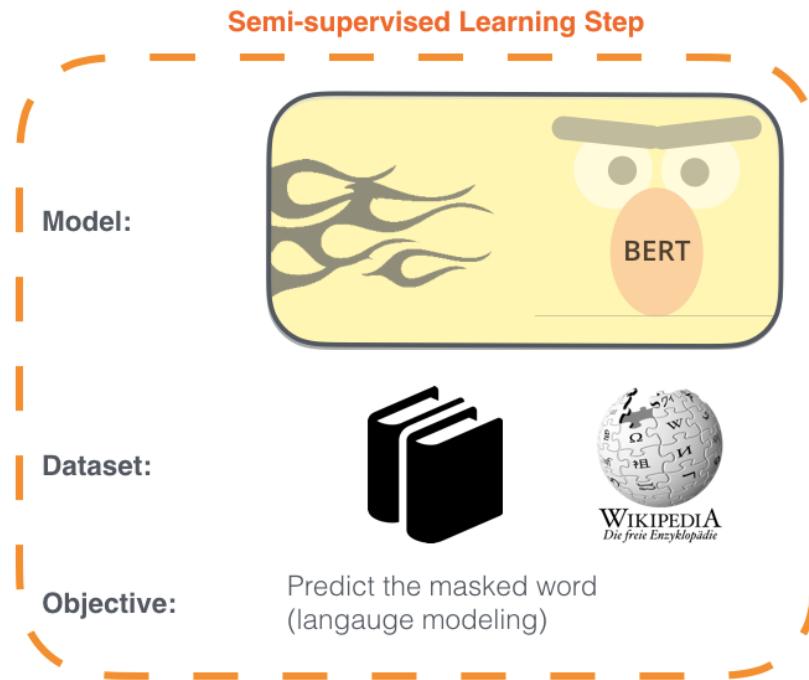


BERT (Bidirectional Encoder Representations from Transformers)

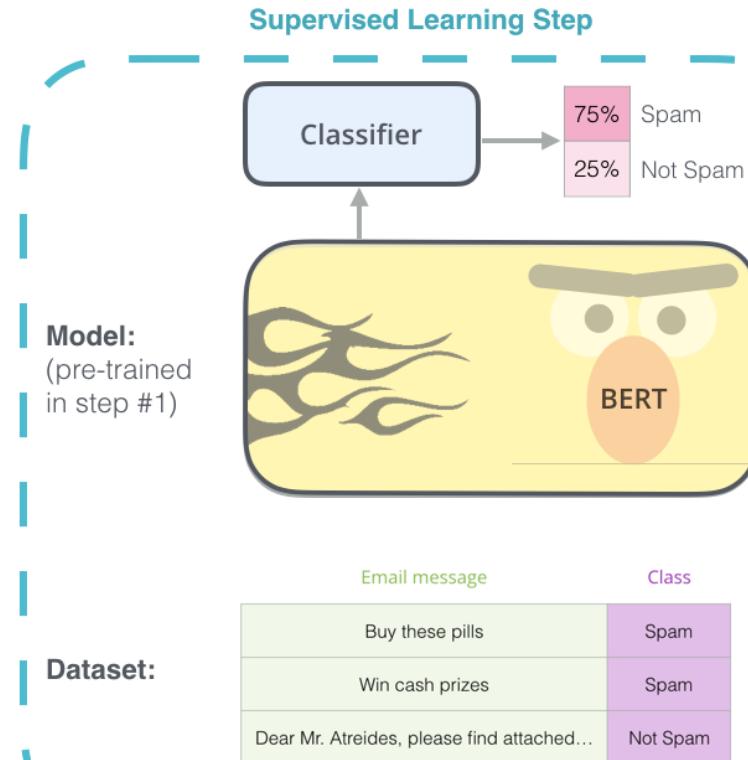
BERT

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

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

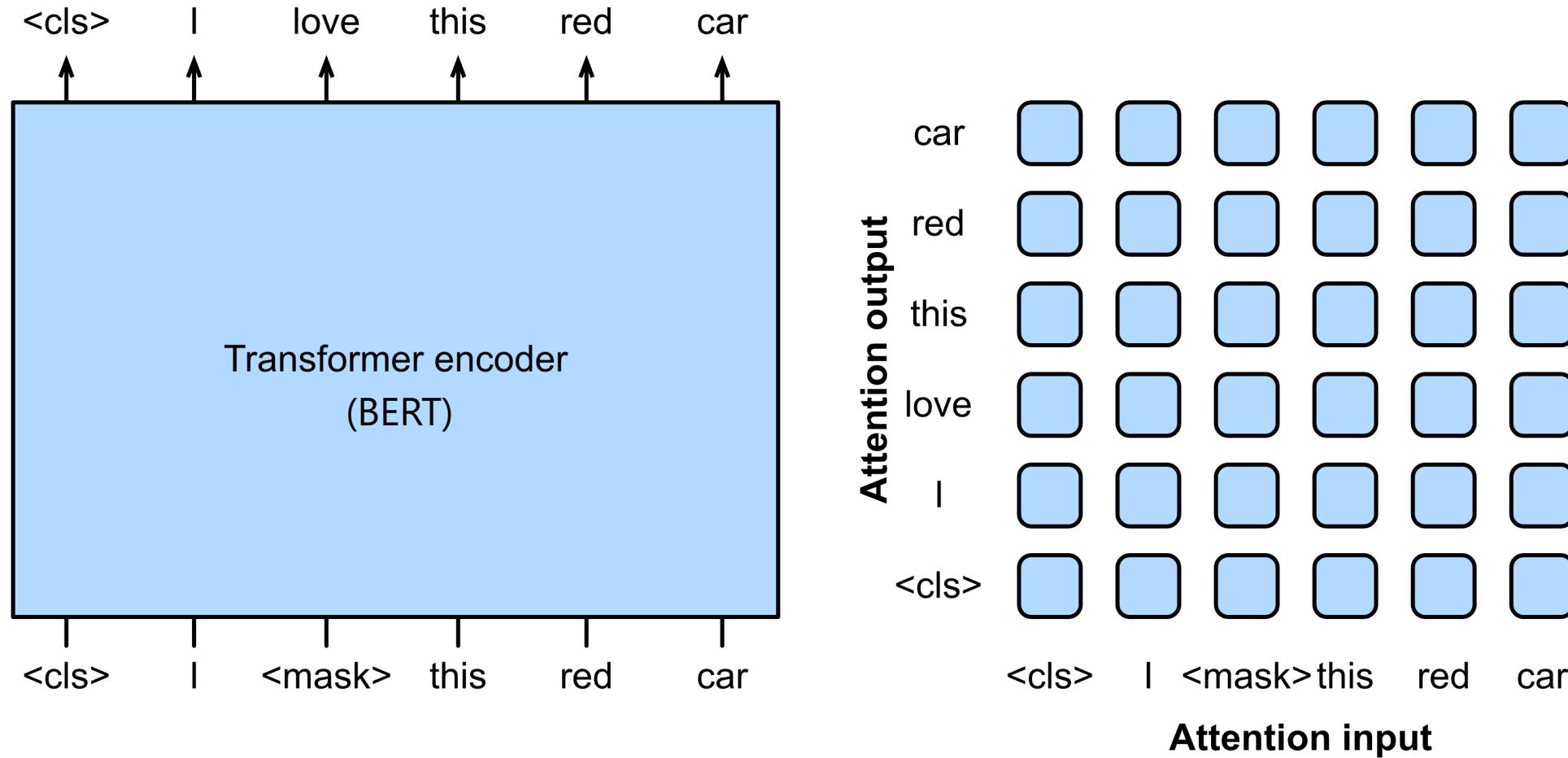


2 - Supervised training on a specific task with a labeled dataset.



The two steps of how BERT is developed. You can download the model pre-trained in step 1 (trained on un-annotated data), and only worry about fine-tuning it for step 2.

Semi-supervised: Masked Language Model



Semi-supervised: Masked Language Model

- ▶ Выкидываем часть слов (обычно 15%), модель должна их восстановить:
 - 80% меняем на токен <MASK>
 - 10% меняем на случайные
 - 10% не меняем
- ▶ Мы предсказываем пропуски, считаем Loss по всем 15%. Случайные и неизменные токены позволяют не переобучиться под токен <MASK>.

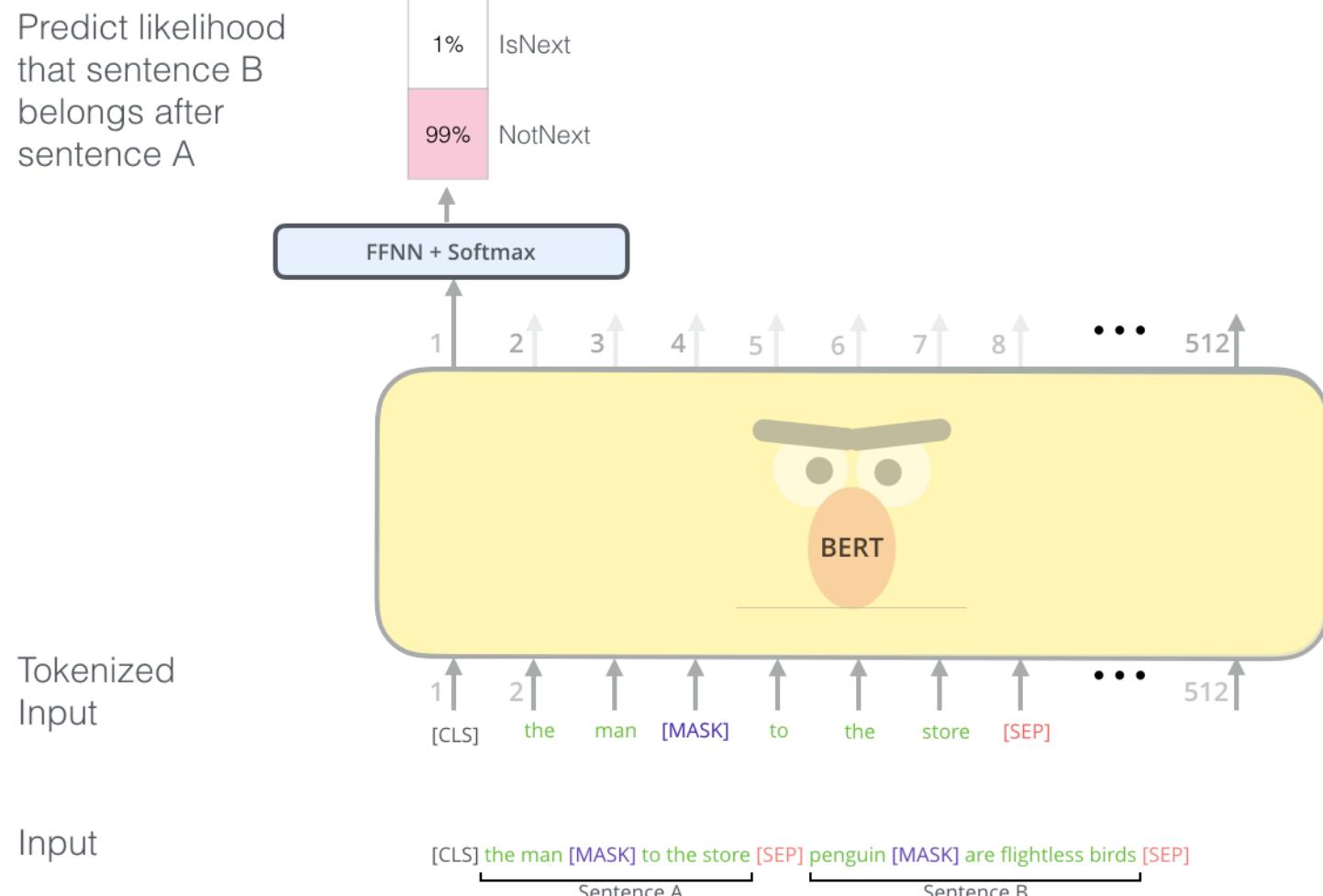
Специальные токены

- ▶ Пример токенезированного предложения:

[CLS] my dog is cute [SEP] he likes play ##ing [SEP]

- ▶ [CLS] — специальный токен, который мы добавляем к началу любого предложения, он несет информацию о всей входной последовательности
- ▶ [SEP] — специальный токен, который говорит, что слева от него первое предложение, а справа — второе
- ▶ ## — специальный токен, обозначающий, что это кусок слова

Semi-supervised: Next Sentence Prediction

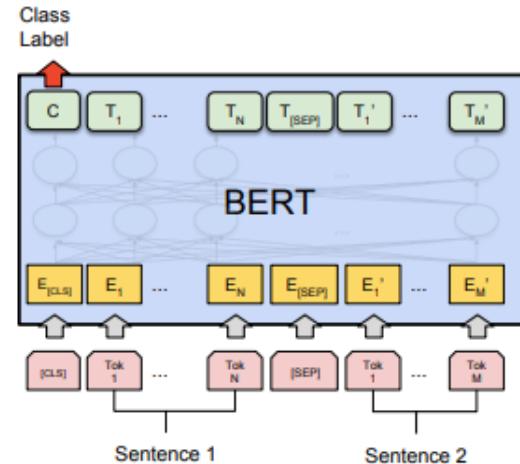


The second task BERT is pre-trained on is a two-sentence classification task. The tokenization is oversimplified in this graphic as BERT actually uses WordPieces as tokens rather than words --- so some words are broken down into smaller chunks.

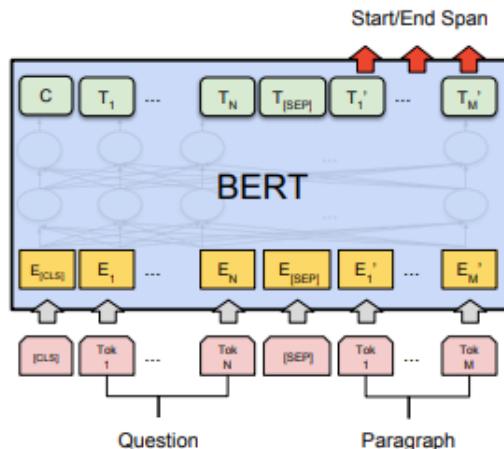
Предобученный BERT

- ▶ BERT обучают решать две задачи: Masked Language Model и Next Sentence Prediction
- ▶ В Masked Language Model модель учится заполнять пропущенные слова. В результате модель умеет давать эмбеддинги слов и предложение, генерировать текст
- ▶ В Next Sentence Prediction модель учит концепцию предложения и отношений между ними
- ▶ Затем мы можем дообучить BERT под наши конкретные задачи

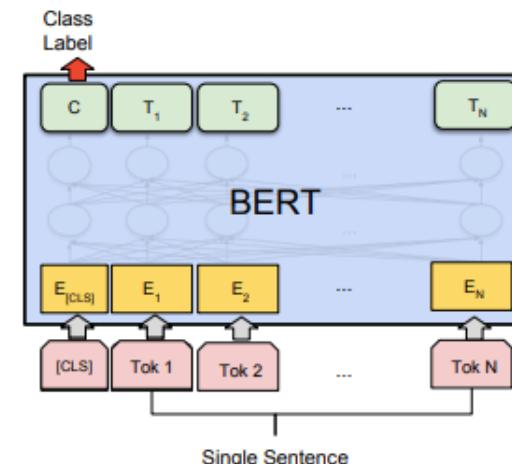
Task specific BERT



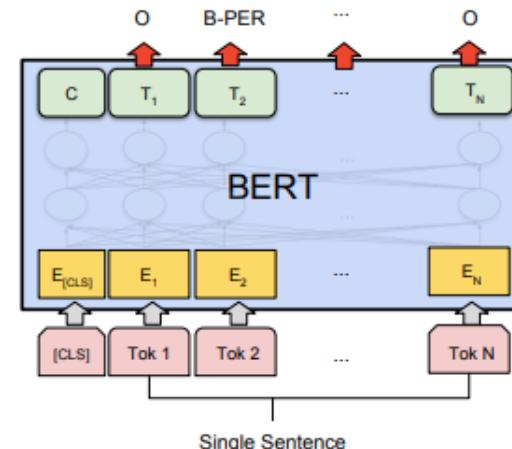
(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG



(c) Question Answering Tasks:
SQuAD v1.1

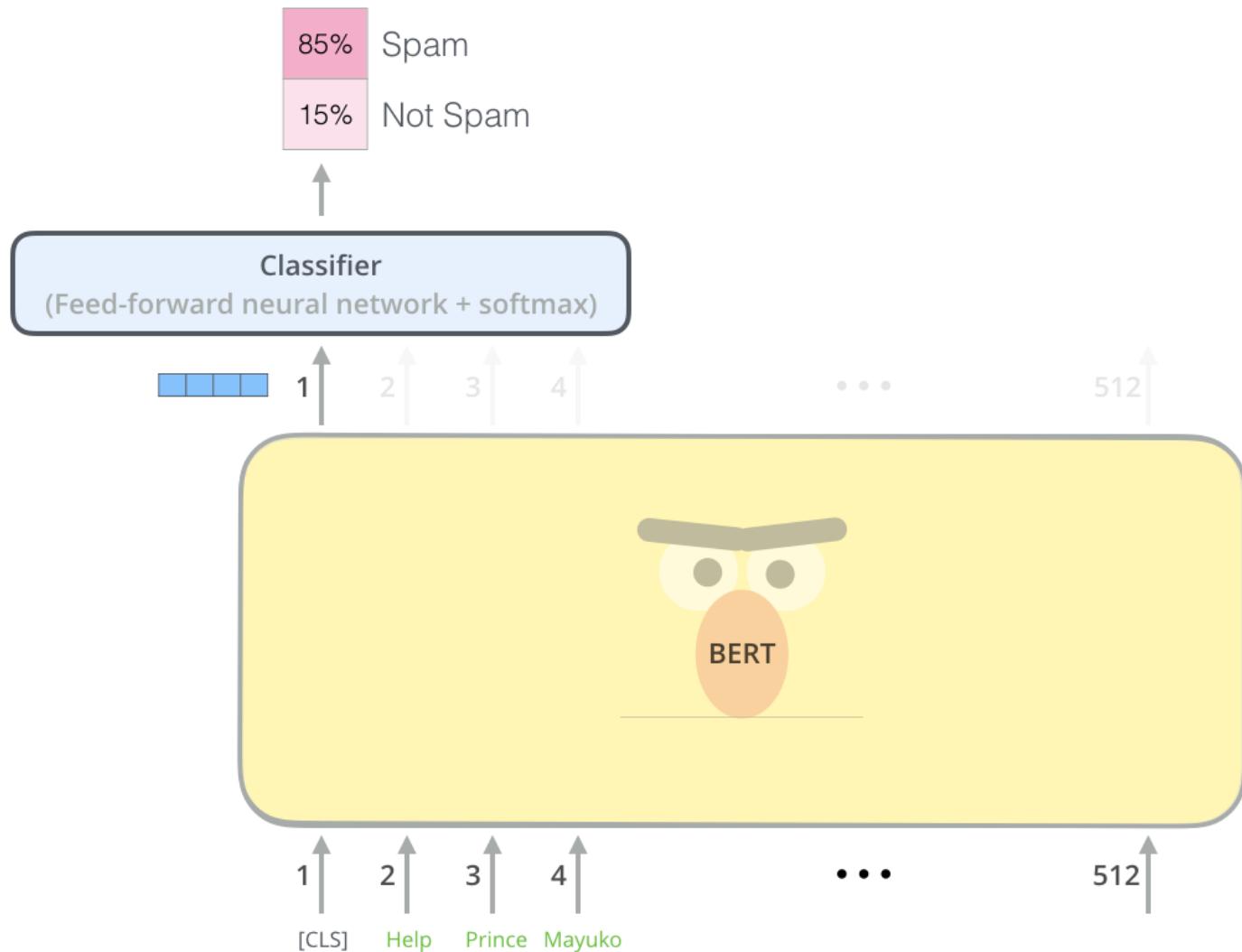


(b) Single Sentence Classification Tasks:
SST-2, CoLA

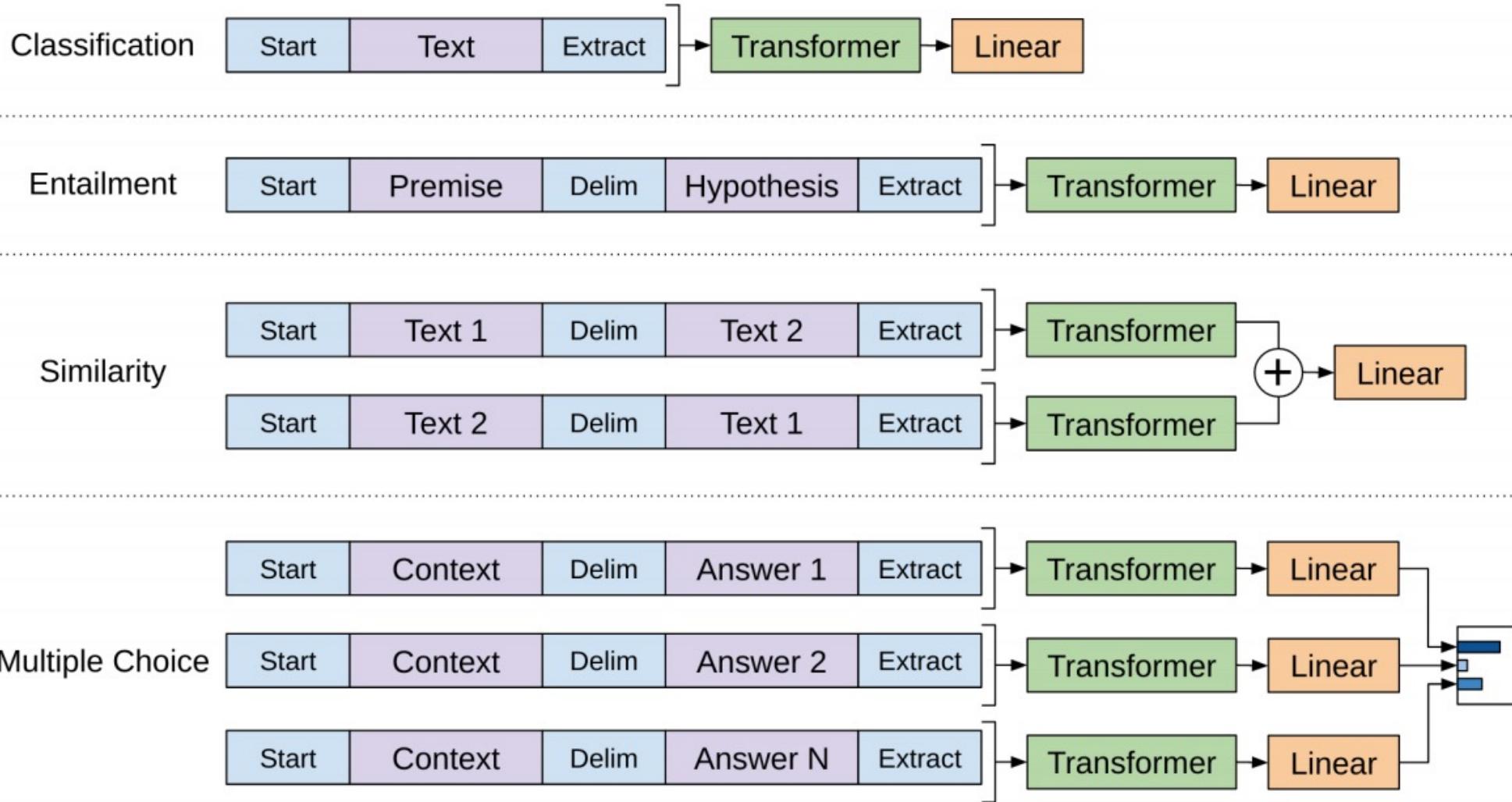
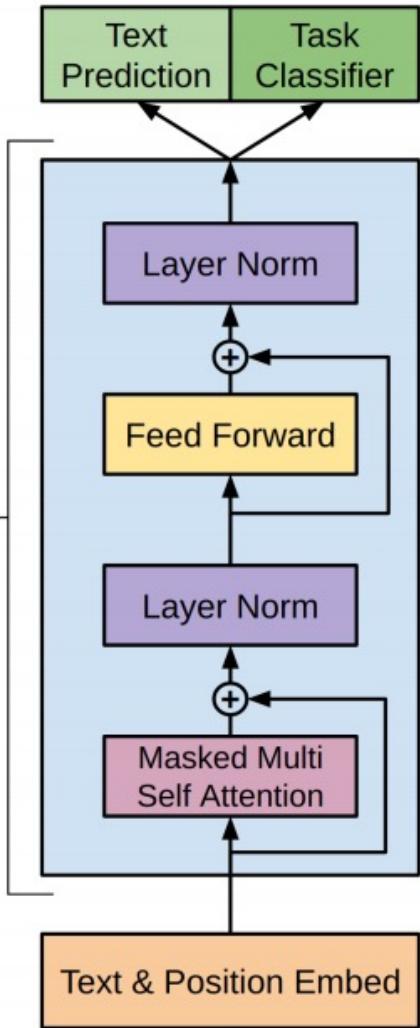


(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Пример: классификация текста



Примеры



Источник: <https://paperswithcode.com/method/gpt>

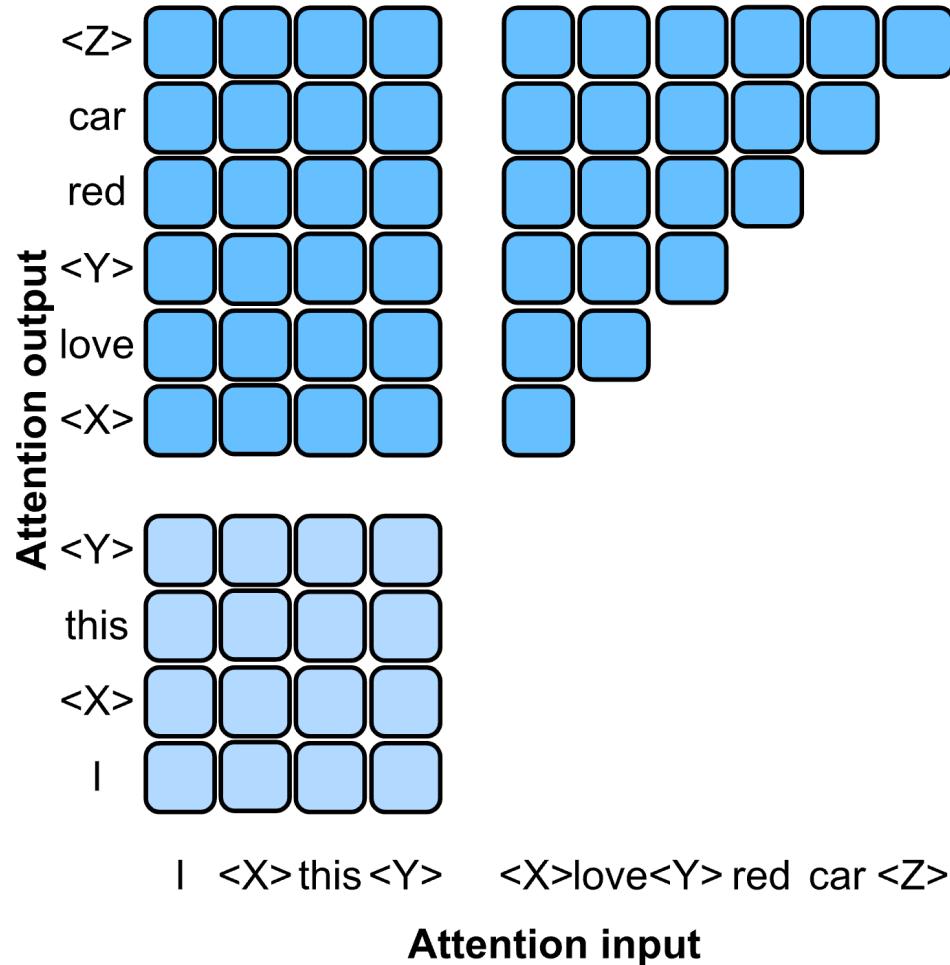
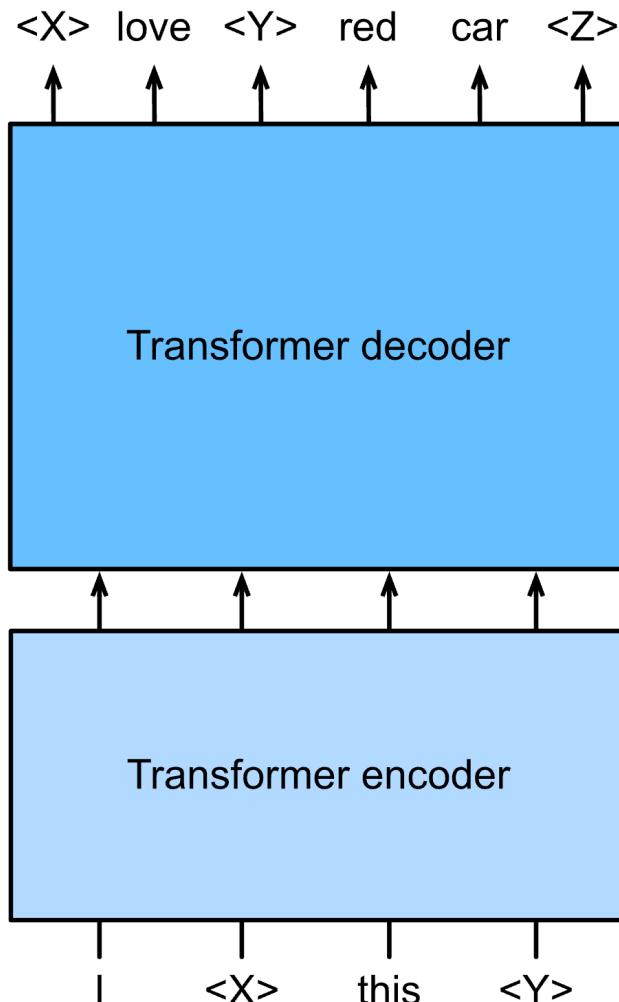
T5 (Text-to-Text Transfer Transformer)



Приложения Т5

- ▶ Модель применяется для задач Text-to-Text
 - Машинный перевод
 - Генерация текста
 - Саммаризация текста
 - Ответы на вопросы по тексту

Обучение T5



Источник: https://d2l.ai/chapter_attention-mechanisms-and-transformers/large-pretraining-transformers.html

Обучение T5

- ▶ Pretraining T5 by predicting consecutive spans.
- ▶ The original sentence is

"I", "love", "this", "red", "car",

where "love" is replaced by a special "<X>" token, and consecutive "red", "car" are replaced by a special "<Y>" token:

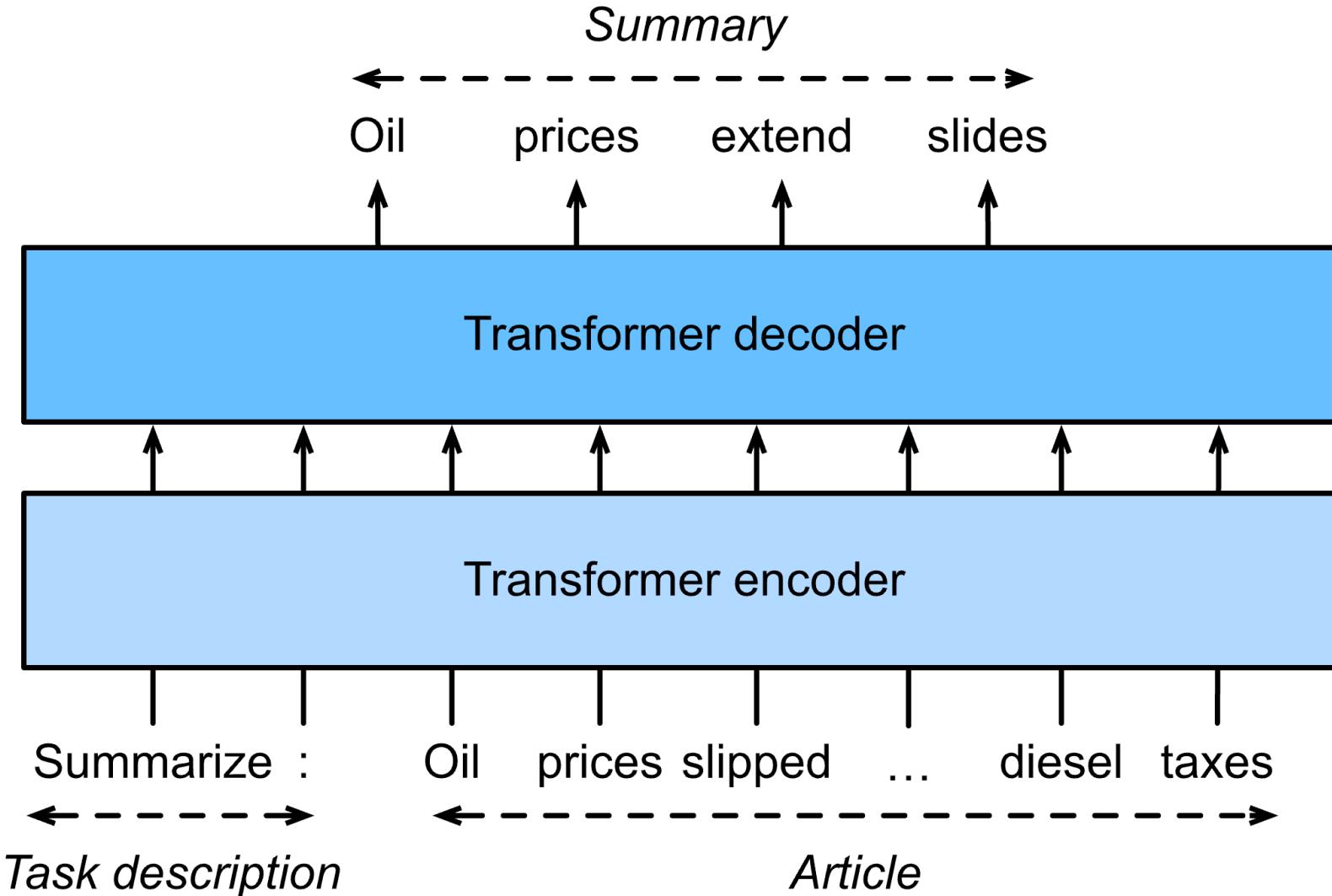
"I", "<X>", "this", "<Y>"

The target sequence ends with a special "<Z>" token.

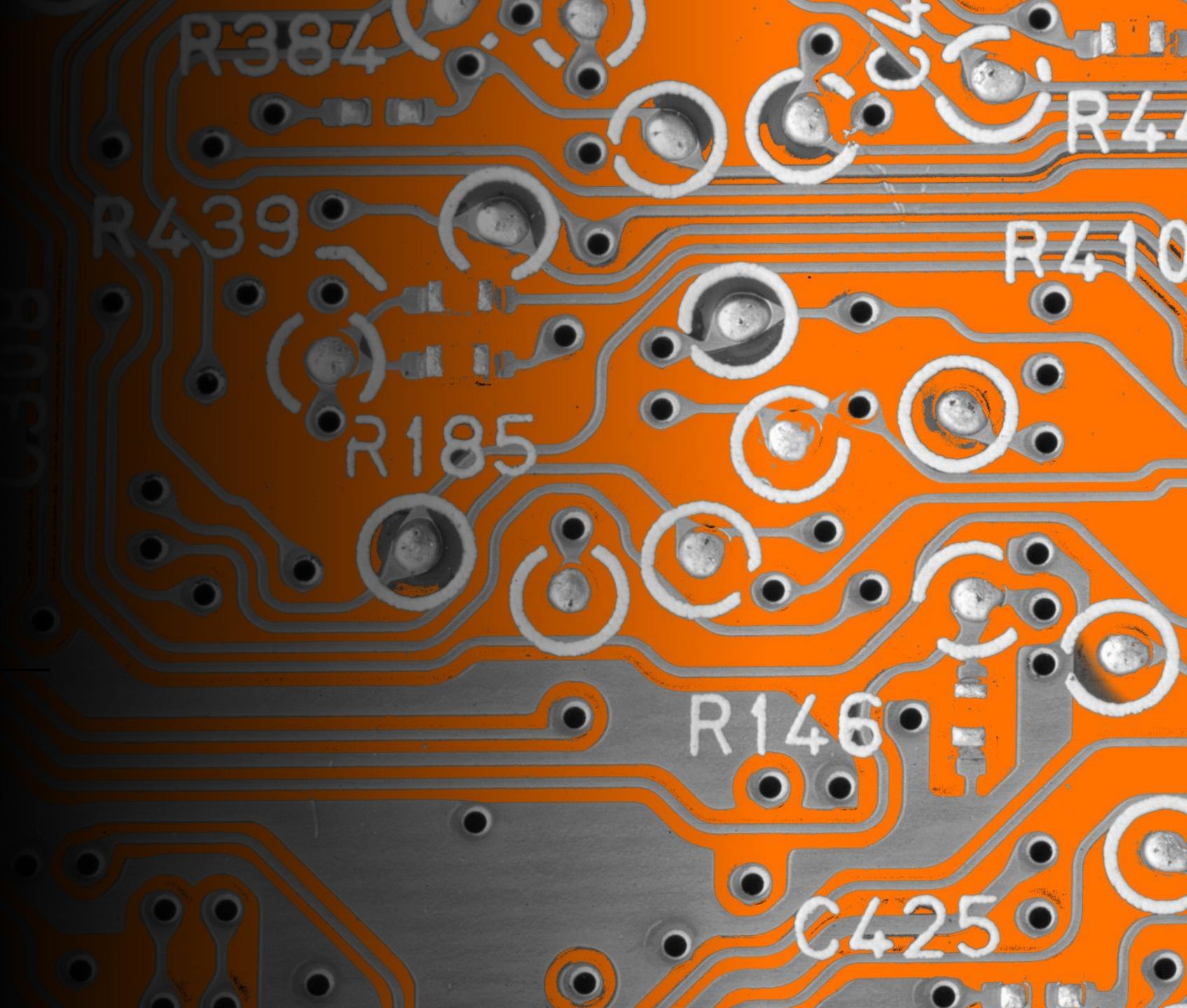
Обучение T5

- ▶ Attention pattern in the Transformer encoder–decoder.
- ▶ In the encoder self-attention (lower square), all input tokens attend to each other;
- ▶ In the encoder–decoder cross-attention (upper rectangle), each target token attends to all input tokens;
- ▶ In the decoder self-attention (upper triangle), each target token attends to present and past target tokens only (causal).

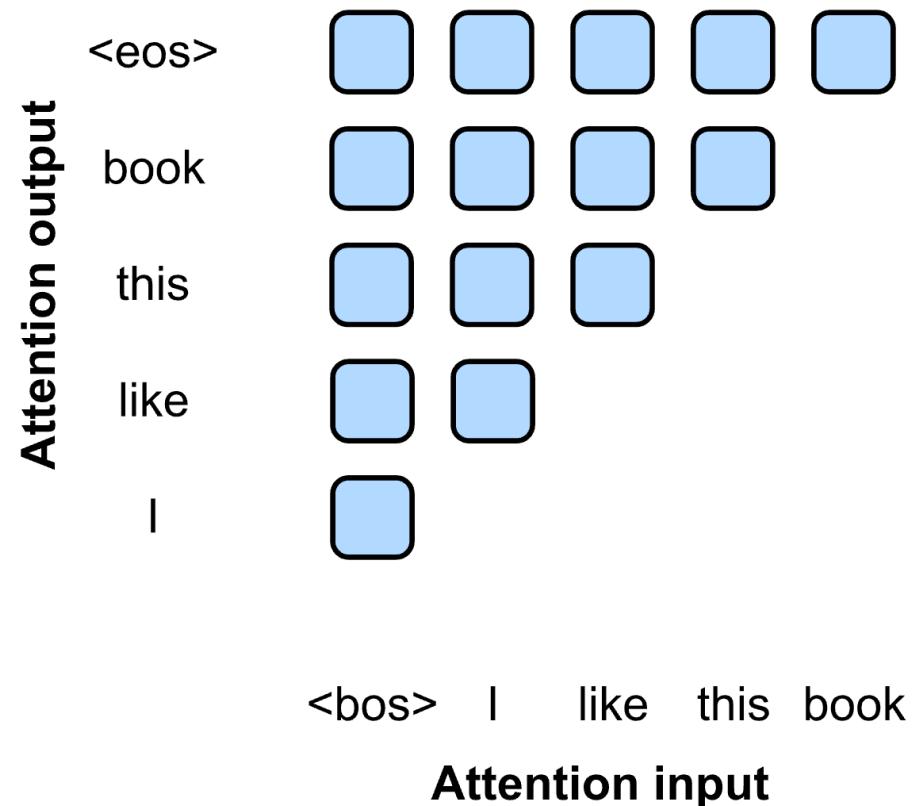
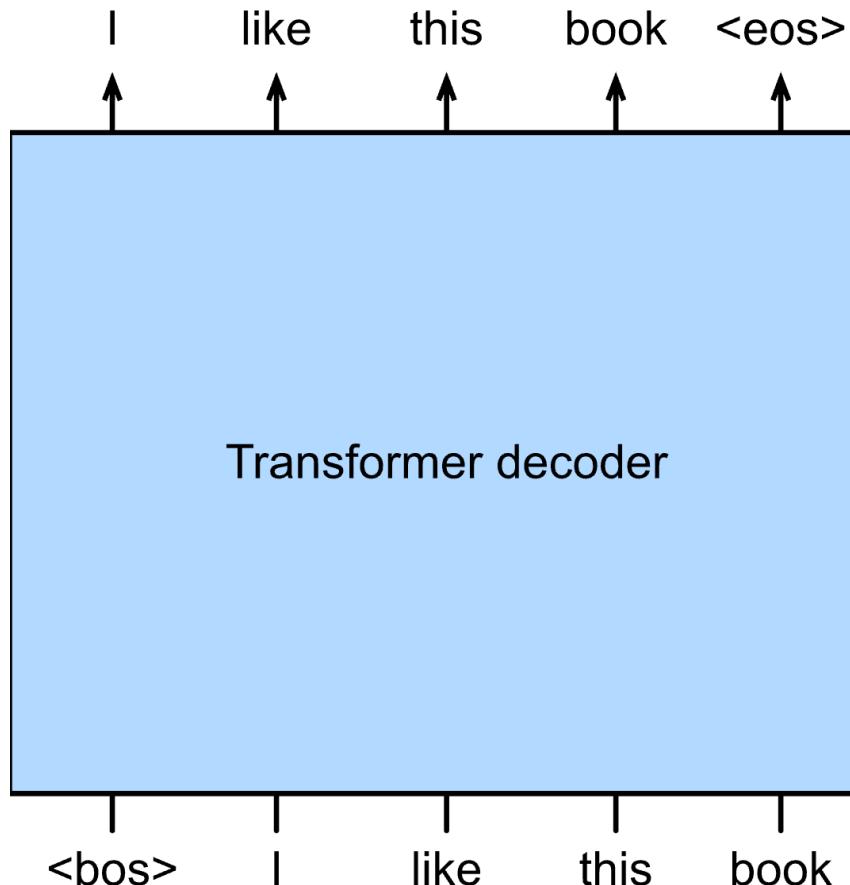
Дообучение Т5



GPT
(Generative
Pre-trained
Transformer)



GPT

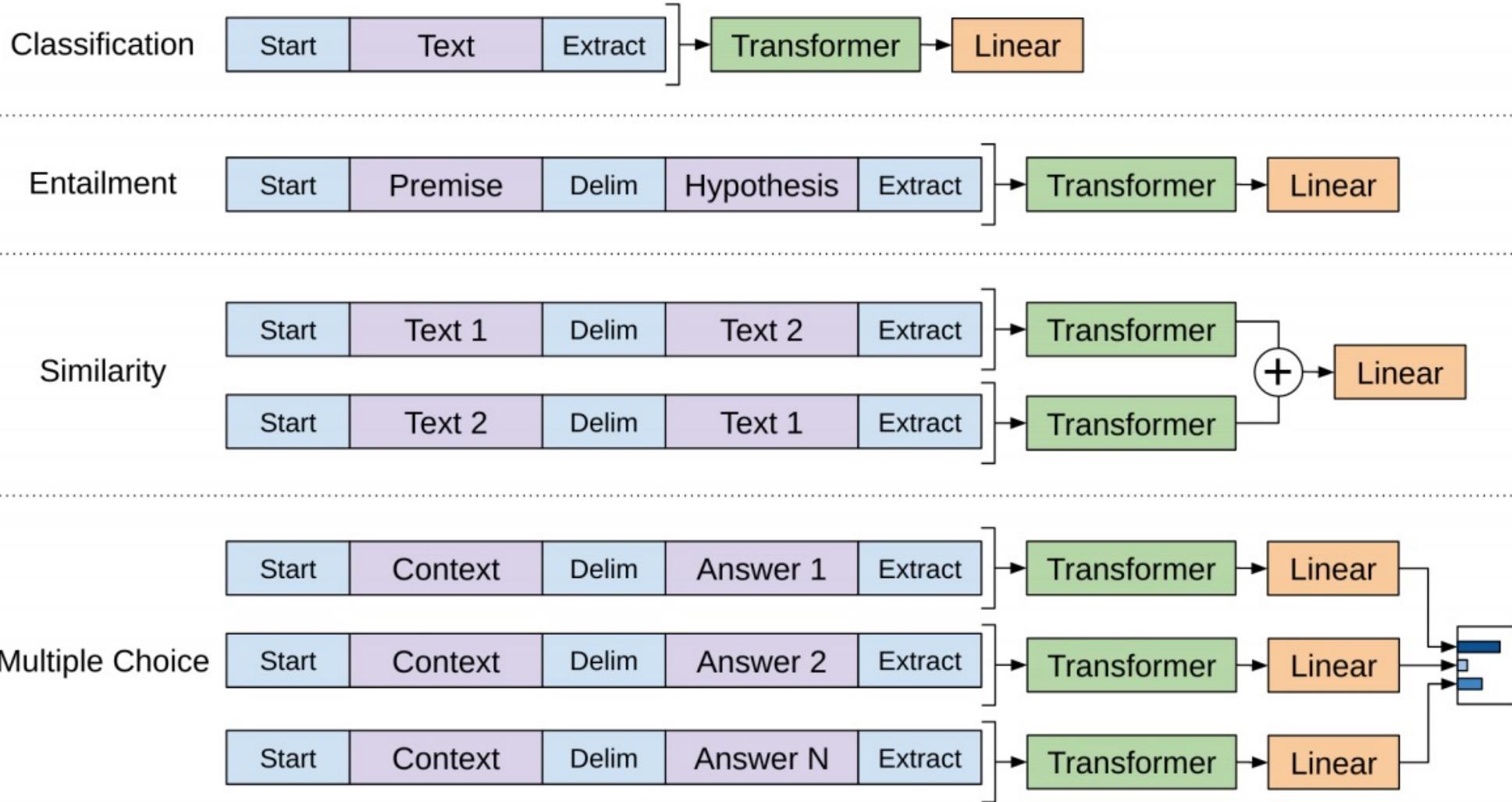
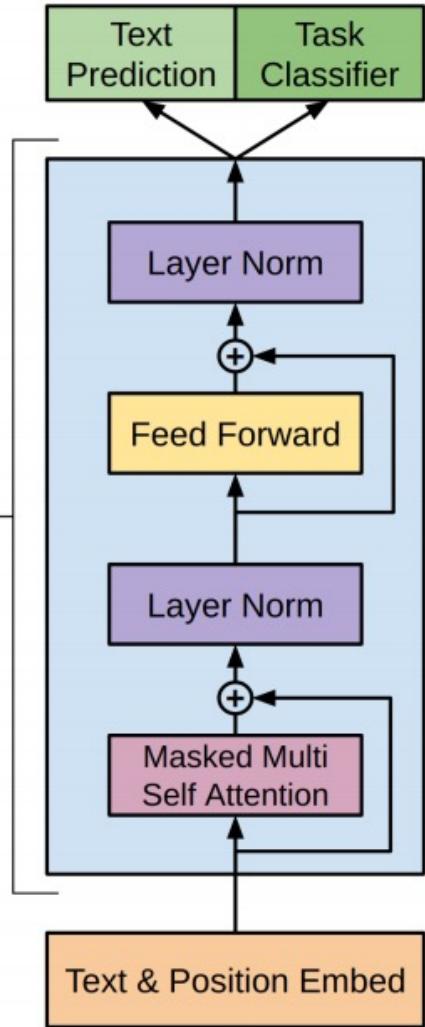


Источник: https://d2l.ai/chapter_attention-mechanisms-and-transformers/large-pretraining-transformers.html

Обучение GPT

- ▶ Обучаем модель предсказывать на один токен вперед
- ▶ Выход – это вход, смещенный на 1 токен
- ▶ «<eos>» - специальный токен начала предложения
- ▶ «<bos>» - специальный токен конца предложения
- ▶ При подсчете attention смотрим только на предыдущие токены (не знаем будущее)

Дообучение



Источник: <https://paperswithcode.com/method/gpt>

GPT-3 и далее

- ▶ Обучаем одну модель сразу решать несколько задач
- ▶ Для этого на вход подаем описание самой задачи:
 - Перевод с русского на английский
 - Саммаризация
 - Генерация текста на тему
 - И другие

GPT-3 и далее

