

Глубинное обучение

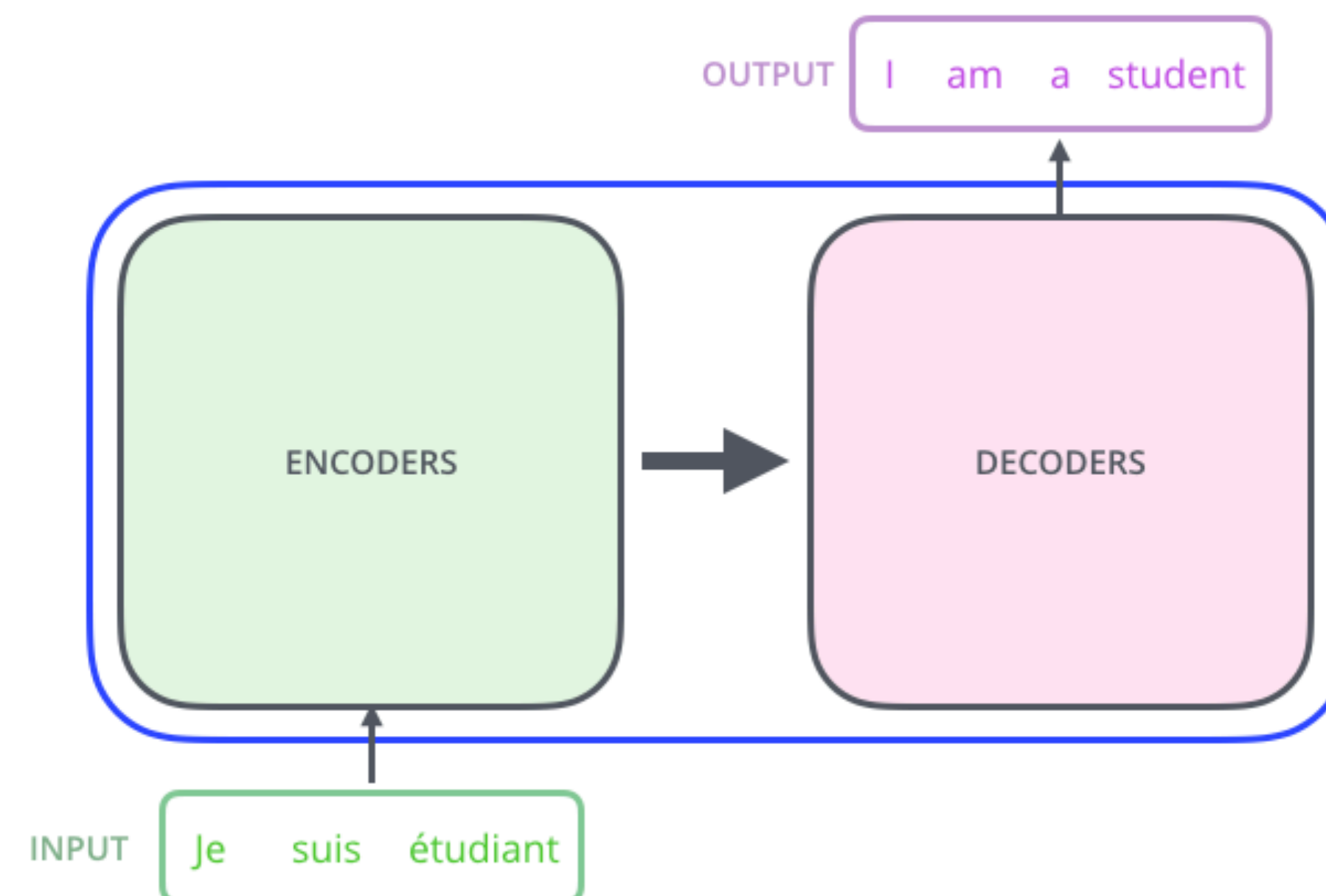
Обработка естественного языка: pre-trained Transformers

Ирина Сапарина

Pre-trained Transformer

Если есть обученный для перевода Transformer, то:

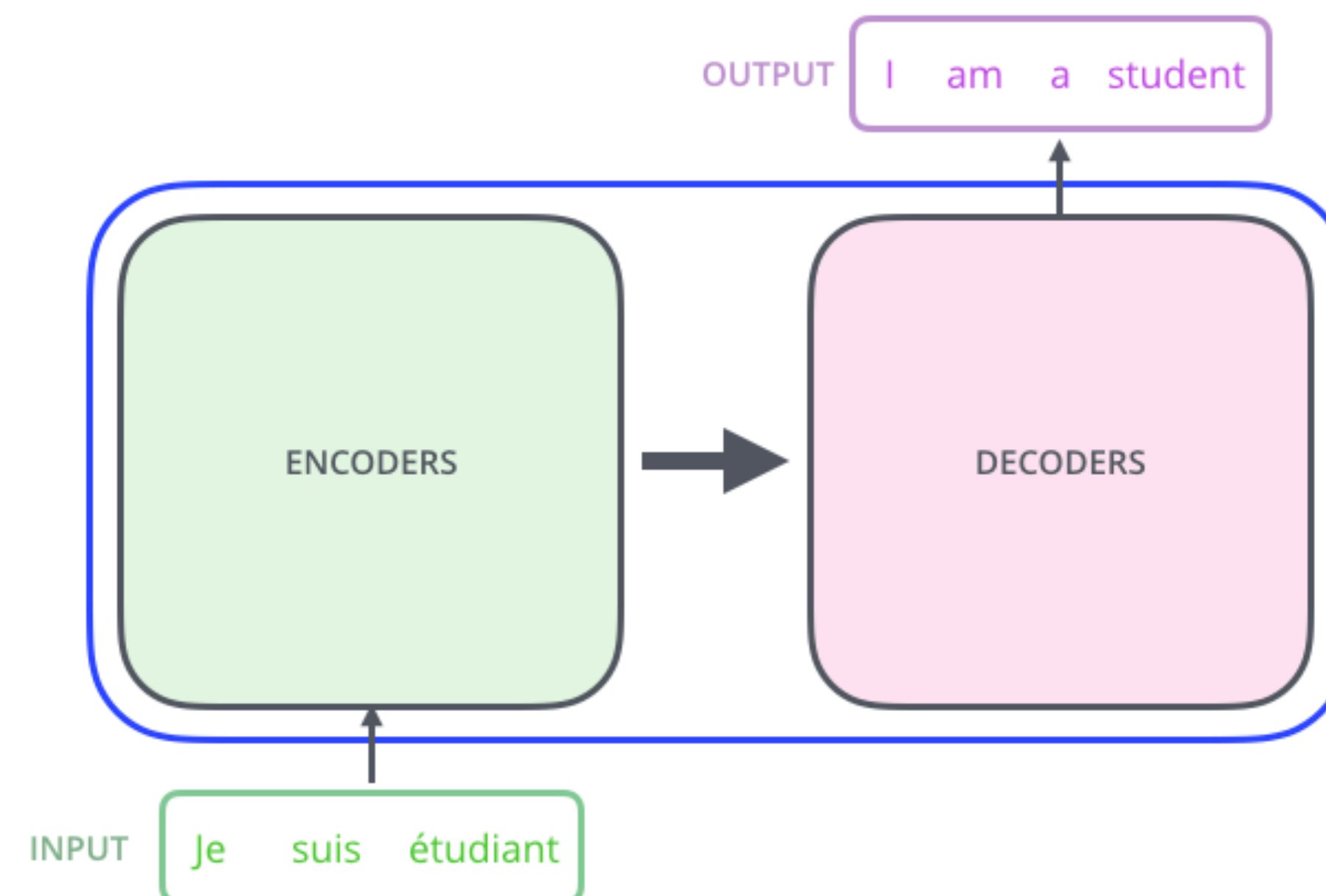
- **Encoder:** выучил хорошие признаки для слов на языке входа
- **Decoder:** выучил хорошие признаки для слов на языке выхода



Pre-trained Transformer

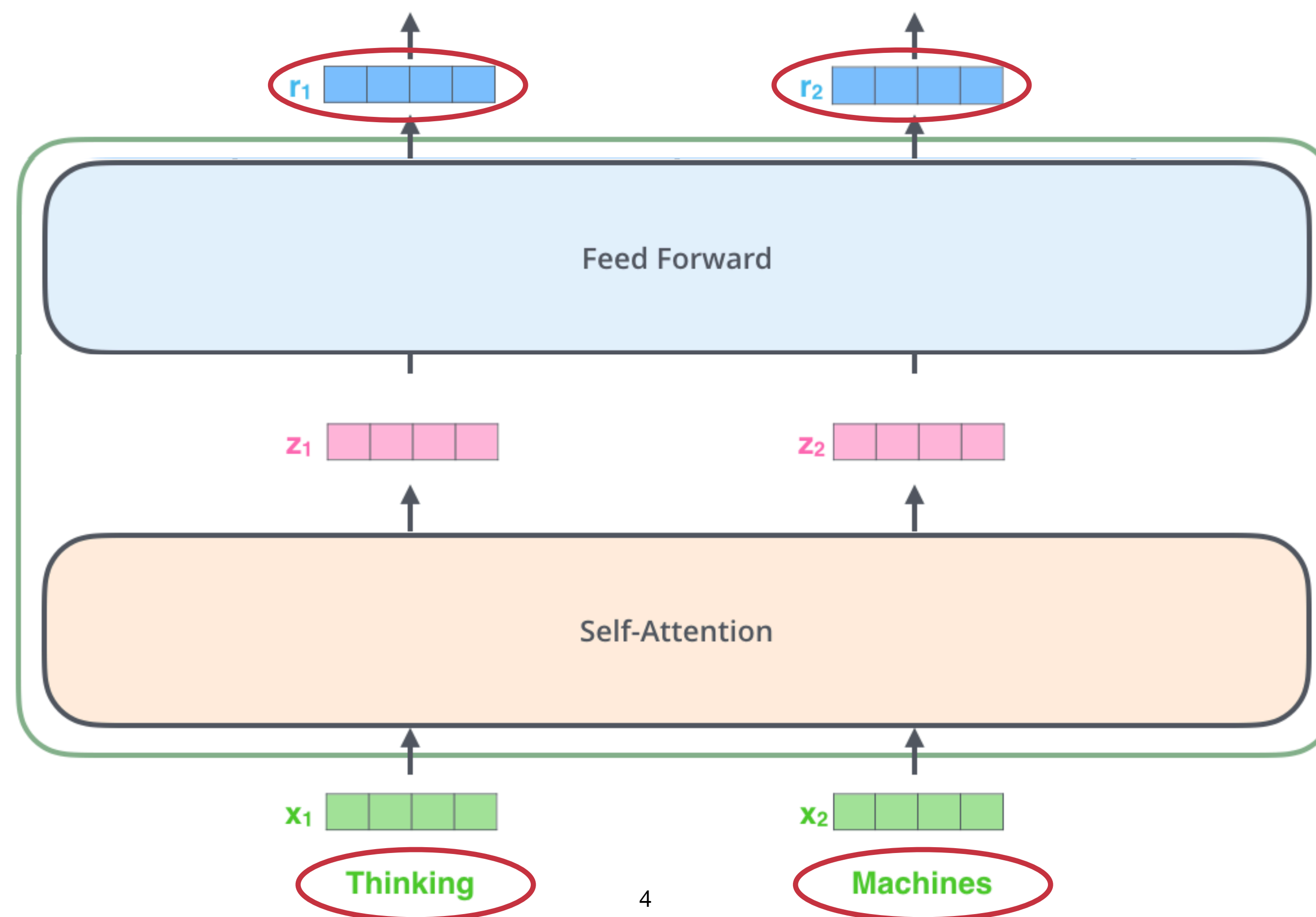
Если есть обученный для перевода Transformer, то:

- **Encoder:** выучил хорошие признаки для слов на языке входа
- **Decoder:** выучил хорошие признаки для слов на языке выхода



Pre-trained Transformer

Для каждого слова Transformer block выдает эмбе́ддинг, зависящий от всего контекста



Pre-trained Transformer

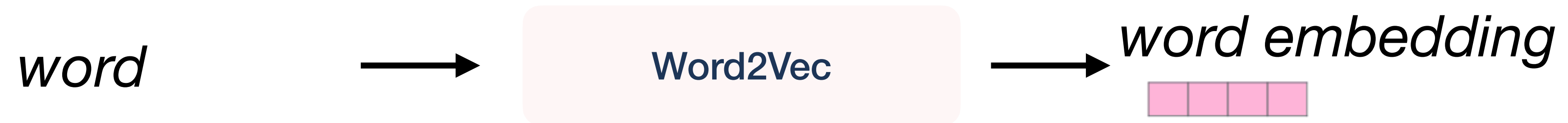
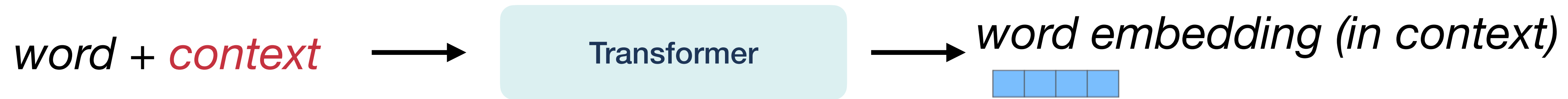
Для каждого слова Transformer block выдает эмбединг, зависящий от всего контекста

- это верно и для train, и для inference

Word2vec: обучаются с учетом контекста, используются независимо

Pre-trained Transformer

Inference:



Pre-trained Transformer

Inference:

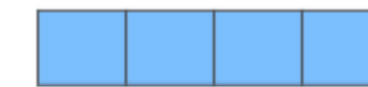
*A **plane** crash*



Transformer



*“plane” embedding
(in context 1)*



*A **plane** surface*



Transformer



*“plane” embedding
(in context 2)*



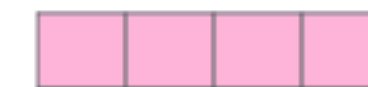
*A **plane** crash /
A **plane** surface*



Word2Vec

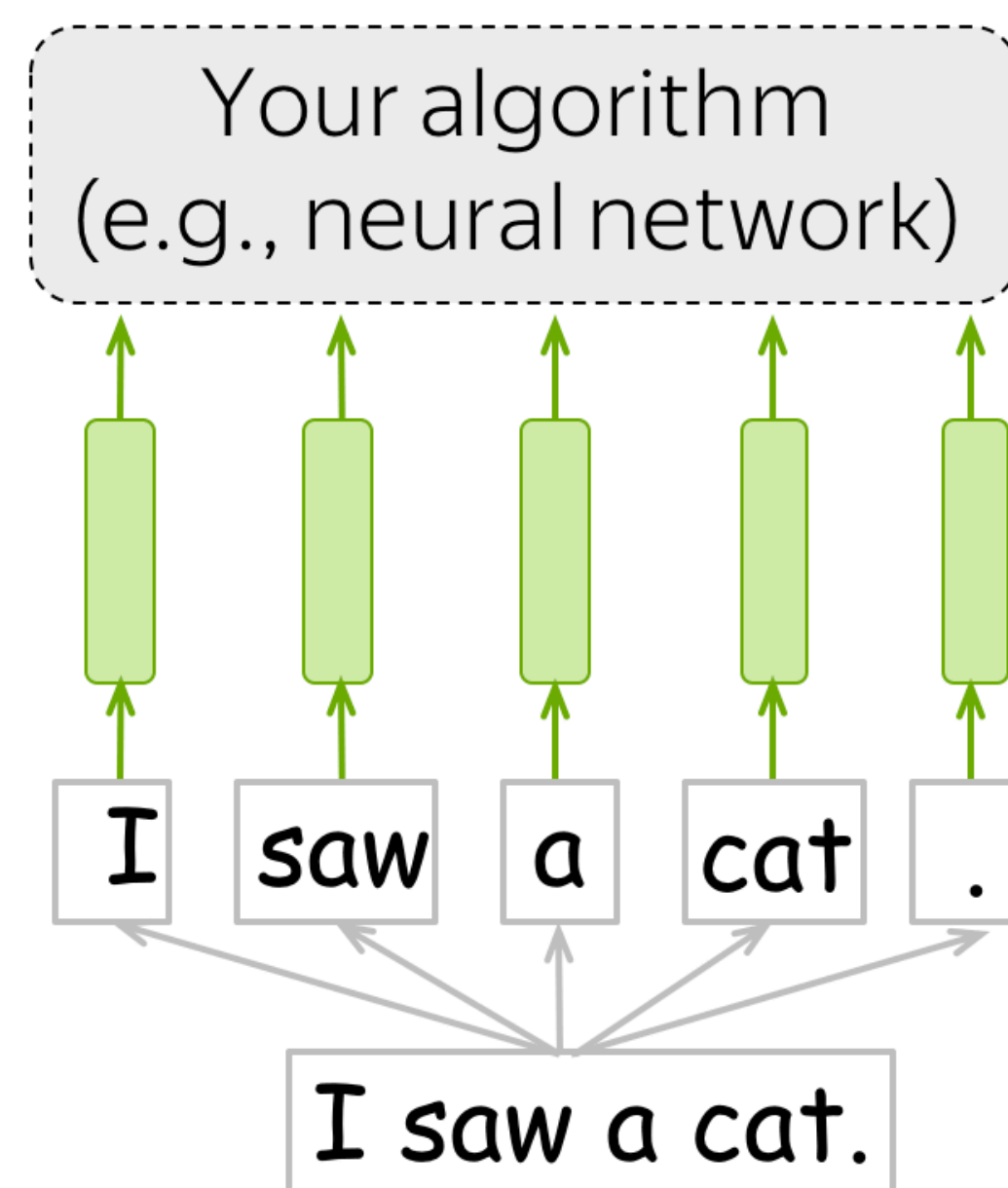


*“plane” embedding
(equal for both contexts)*



Pre-trained Transformer

Как мы используем эмбединги?



Any algorithm for solving a task

Word representation - vector
(input for your model/algorithm)

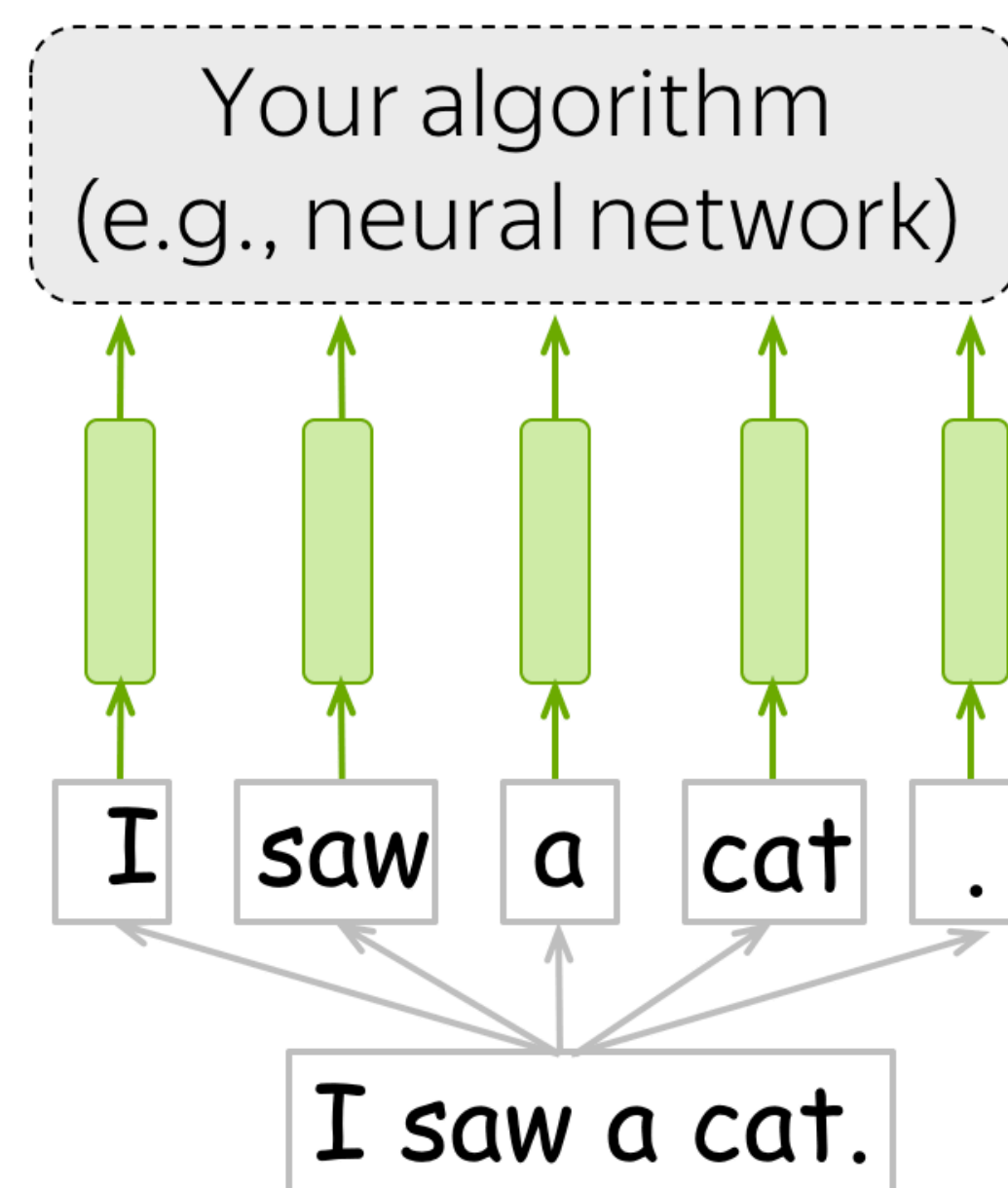
Sequence of tokens

Text (your input)

Pre-trained Transformer

Основная идея: взять предобученные эмбединги (преобученный Transformer) и дообучить на нужную задачу (возможно, с добавлением “головы”)

Pre-train + Fine-tune (Transfer Learning)



Any algorithm for solving a task

Word representation - vector
(input for your model/algorithm)

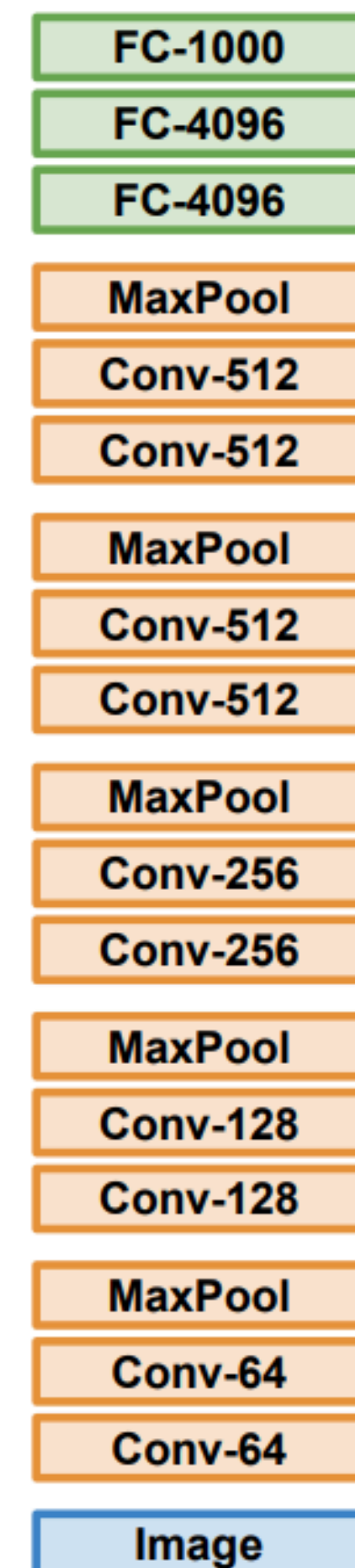
Sequence of tokens

Text (your input)

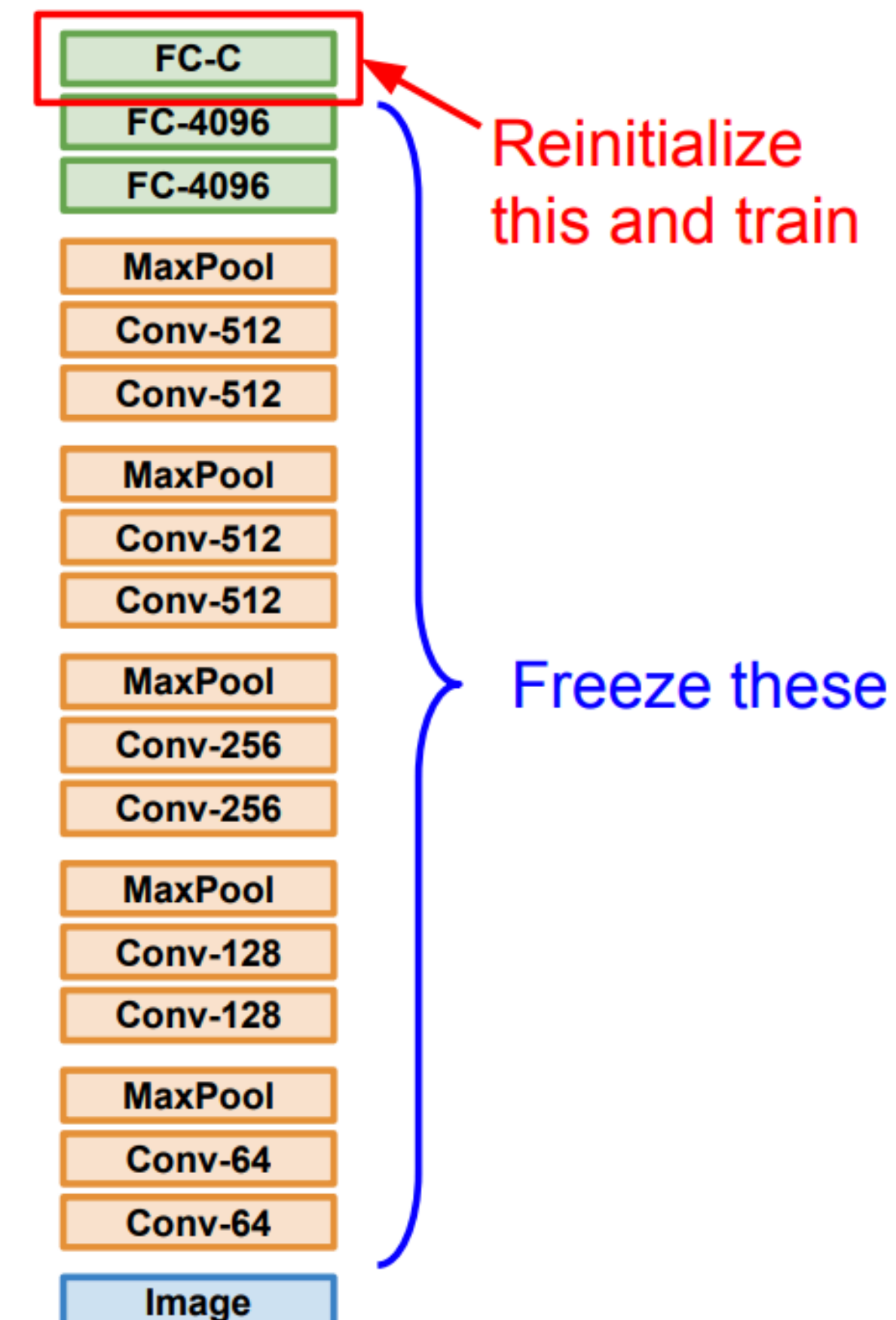
Pre-trained Transformer

Pre-train + Fine-tune (Transfer Learning) в Computer Vision:

1. Train on Imagenet



2. Small Dataset (C classes)

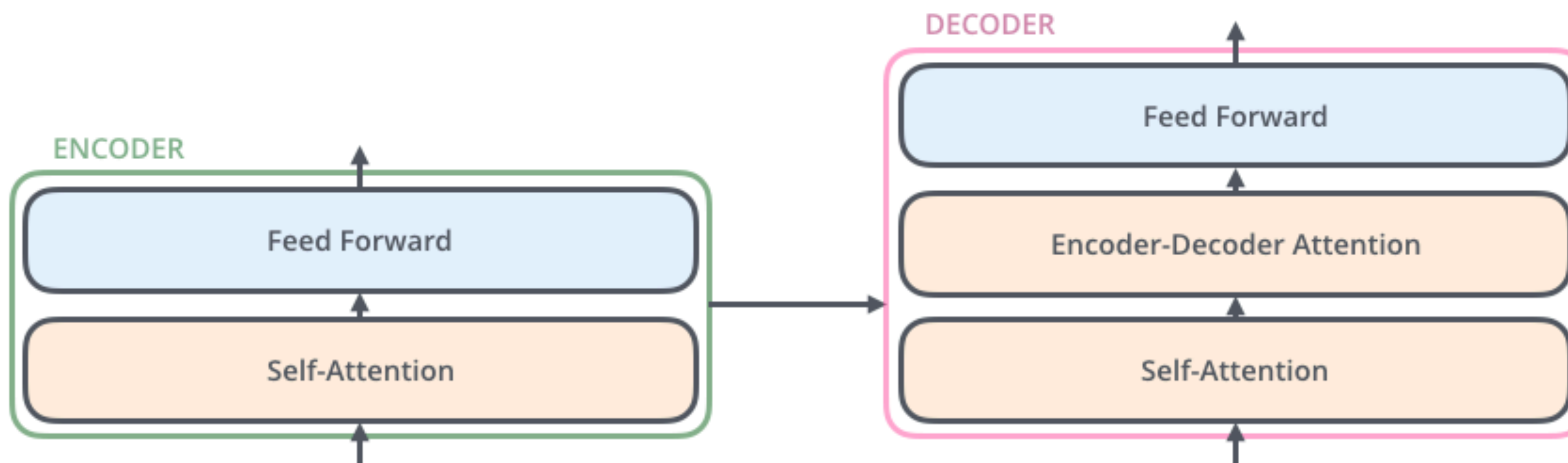


GPT

GPT

Generative Pre-Training

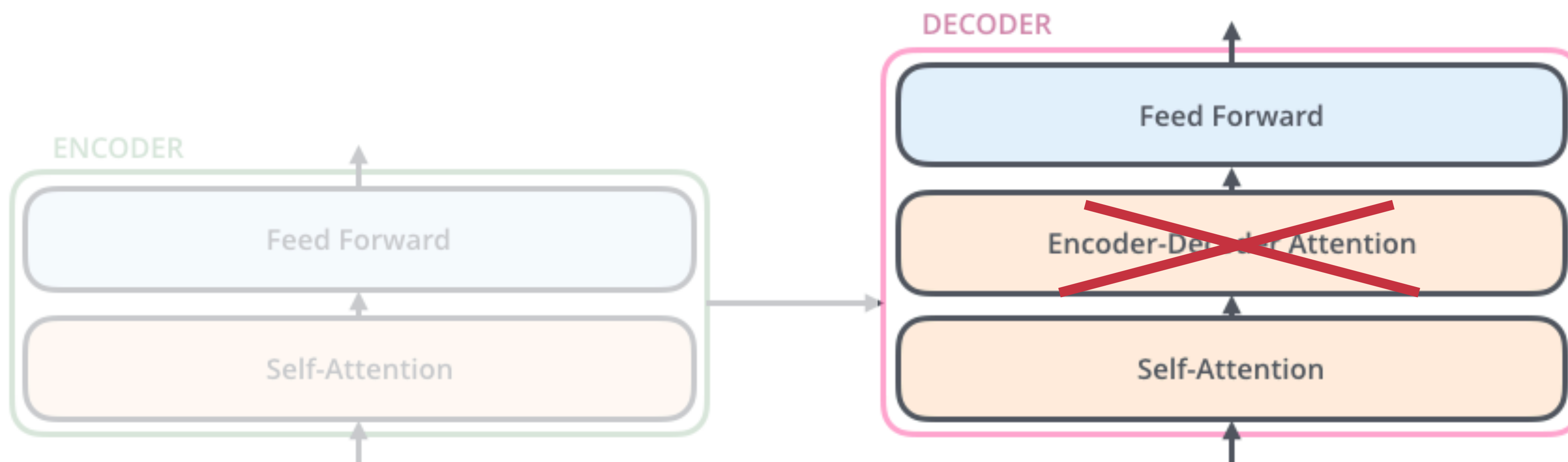
Архитектура: Transformer Decoder



GPT

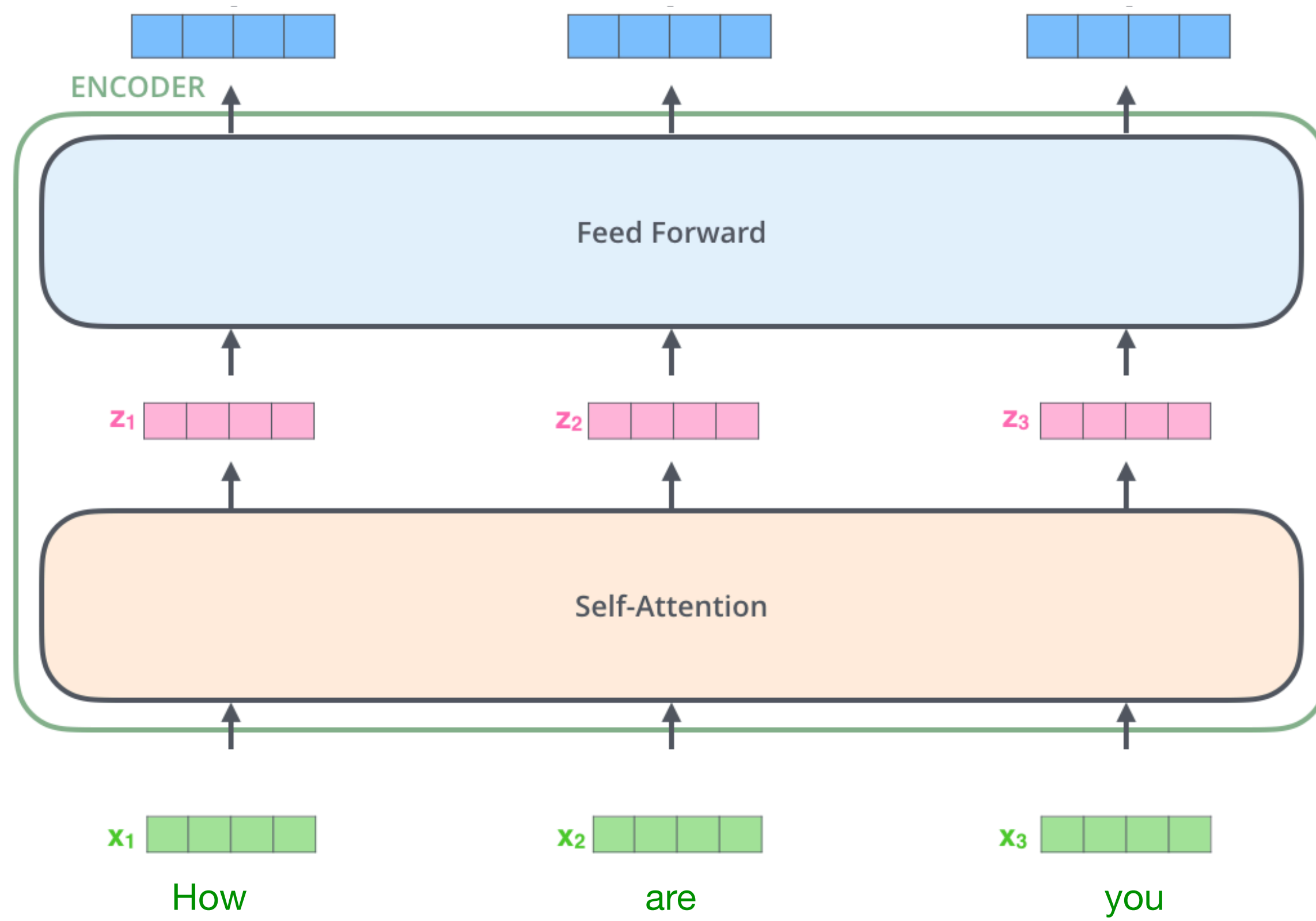
Generative Pre-Training

Архитектура: Transformer Decoder



Encoder vs Decoder

Inference

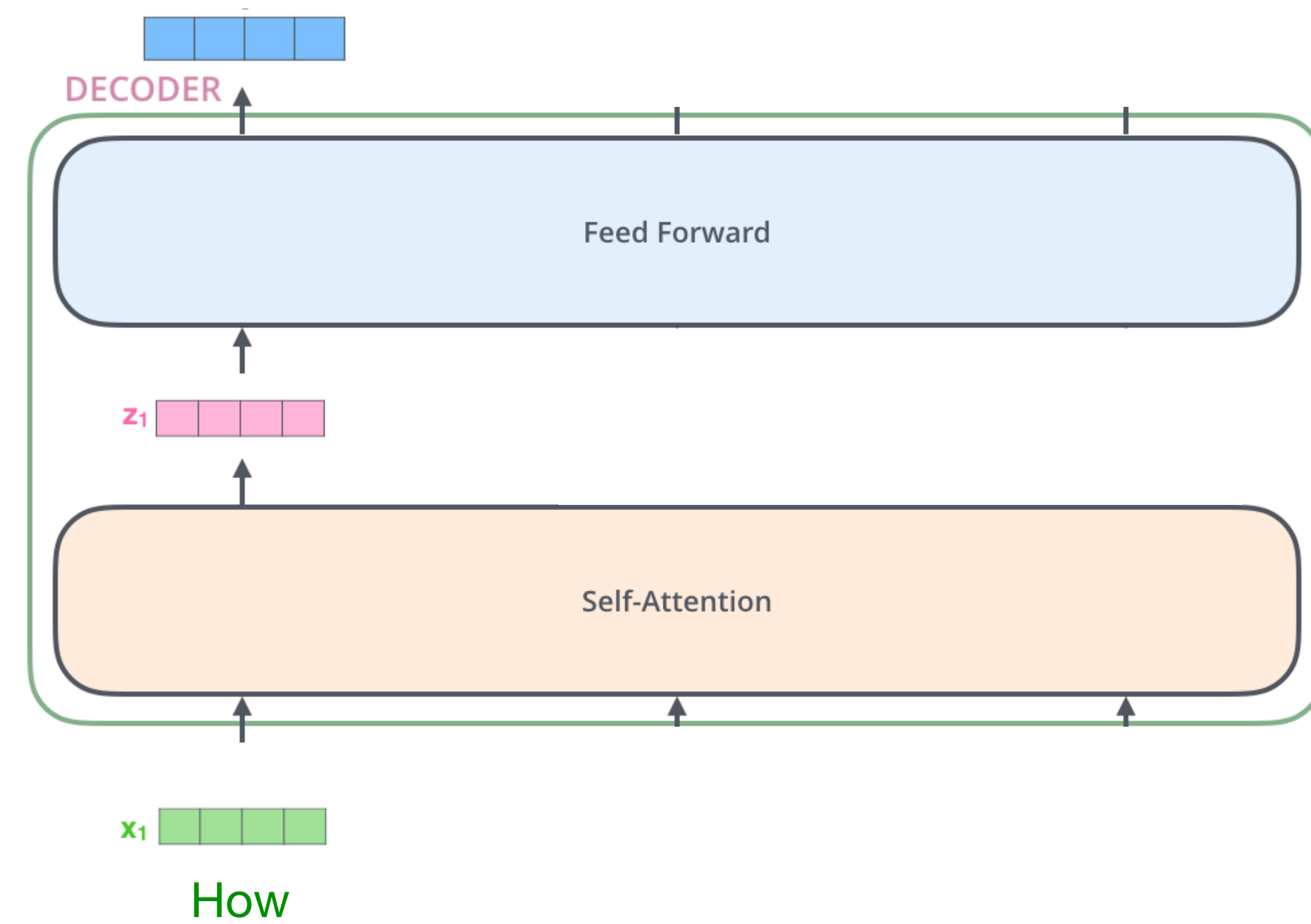
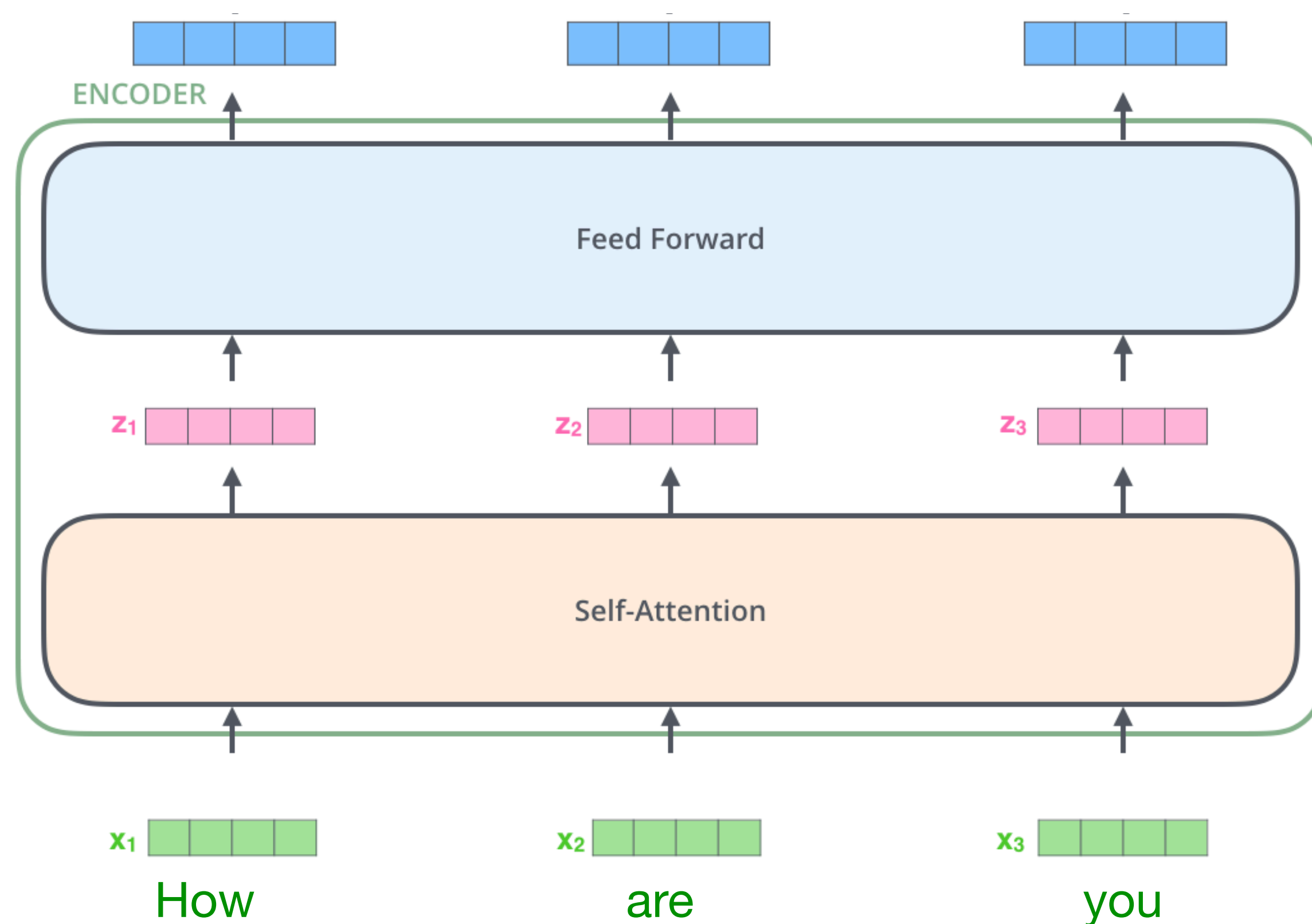


Encoder: одновременно обрабатываем

Encoder vs Decoder

Decoder step 1

Inference



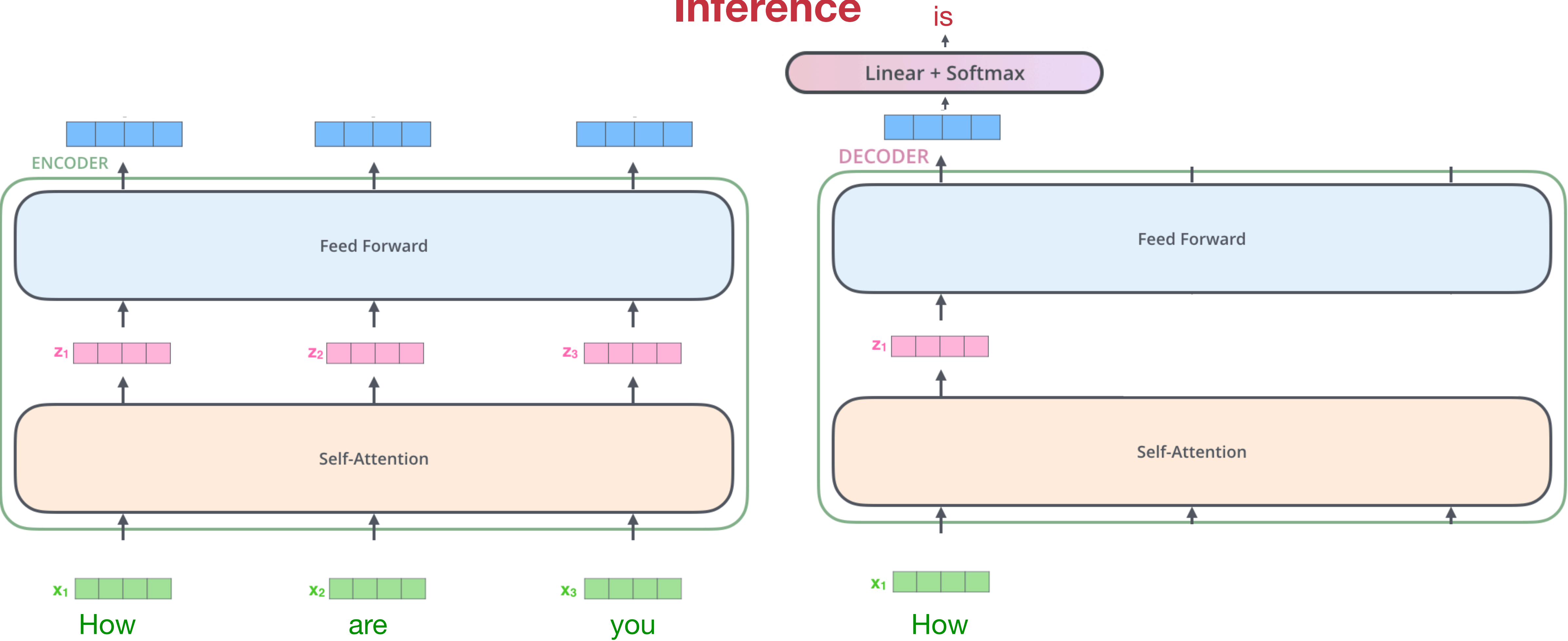
Encoder: одновременно обрабатываем

Decoder: последовательно

Encoder vs Decoder

Decoder step 1

Inference



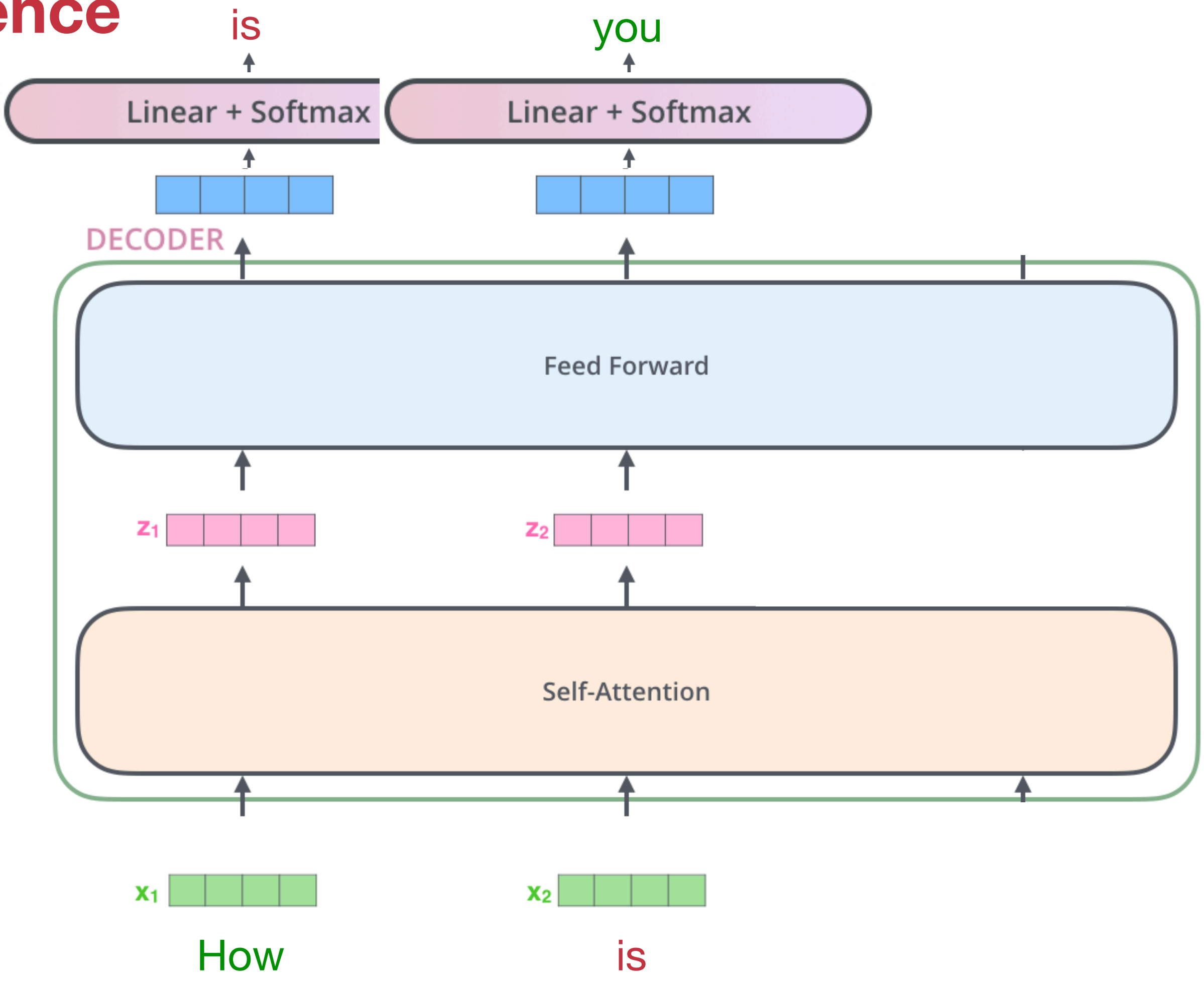
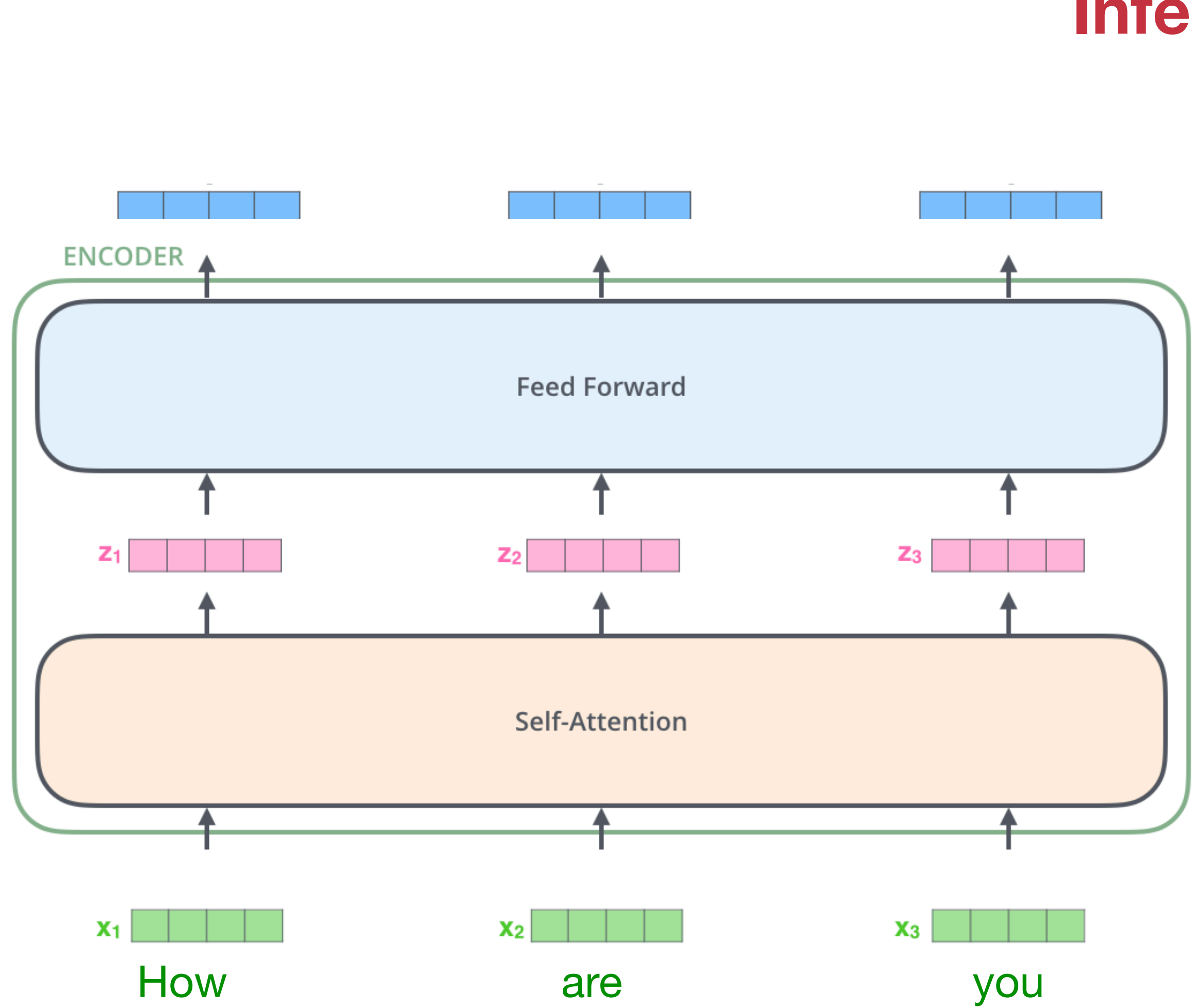
Encoder: одновременно обрабатываем

Decoder: последовательно

Encoder vs Decoder

Decoder step 2

Inference



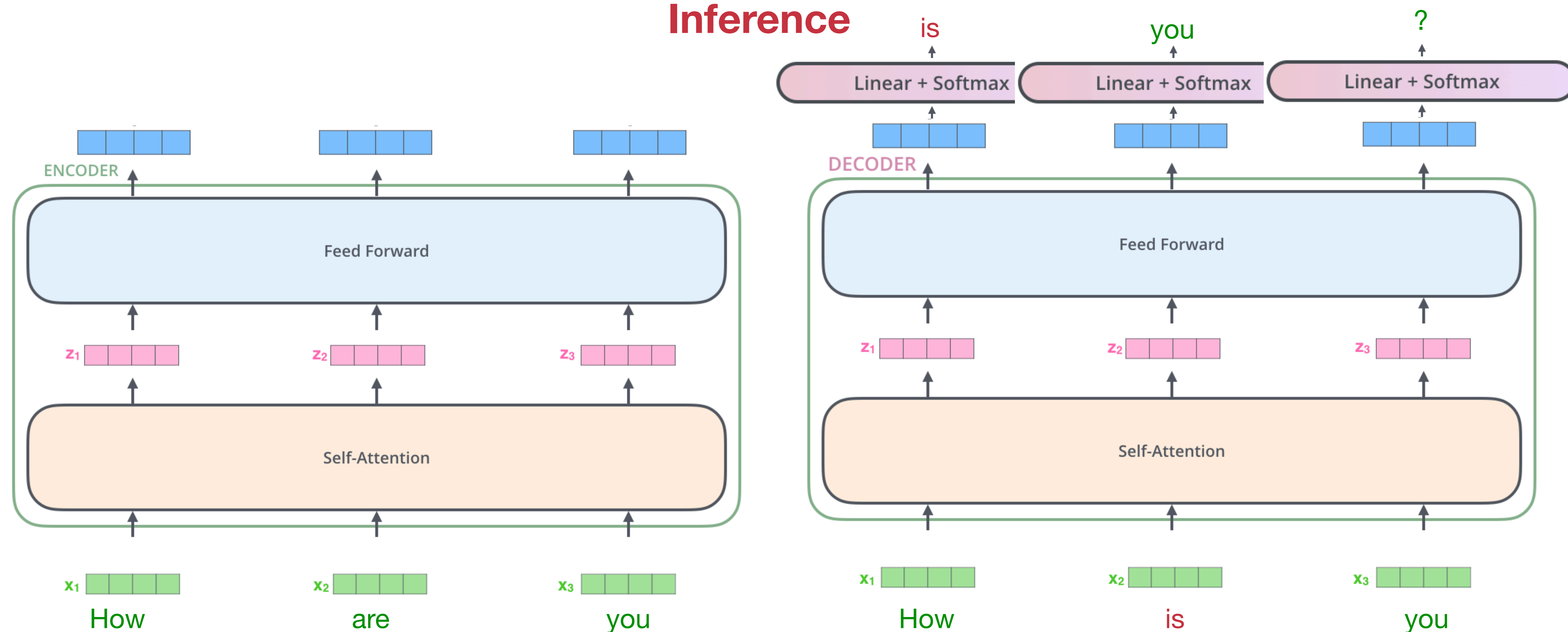
Encoder: одновременно обрабатываем

Decoder: последовательно

Encoder vs Decoder

Decoder step 3

Inference

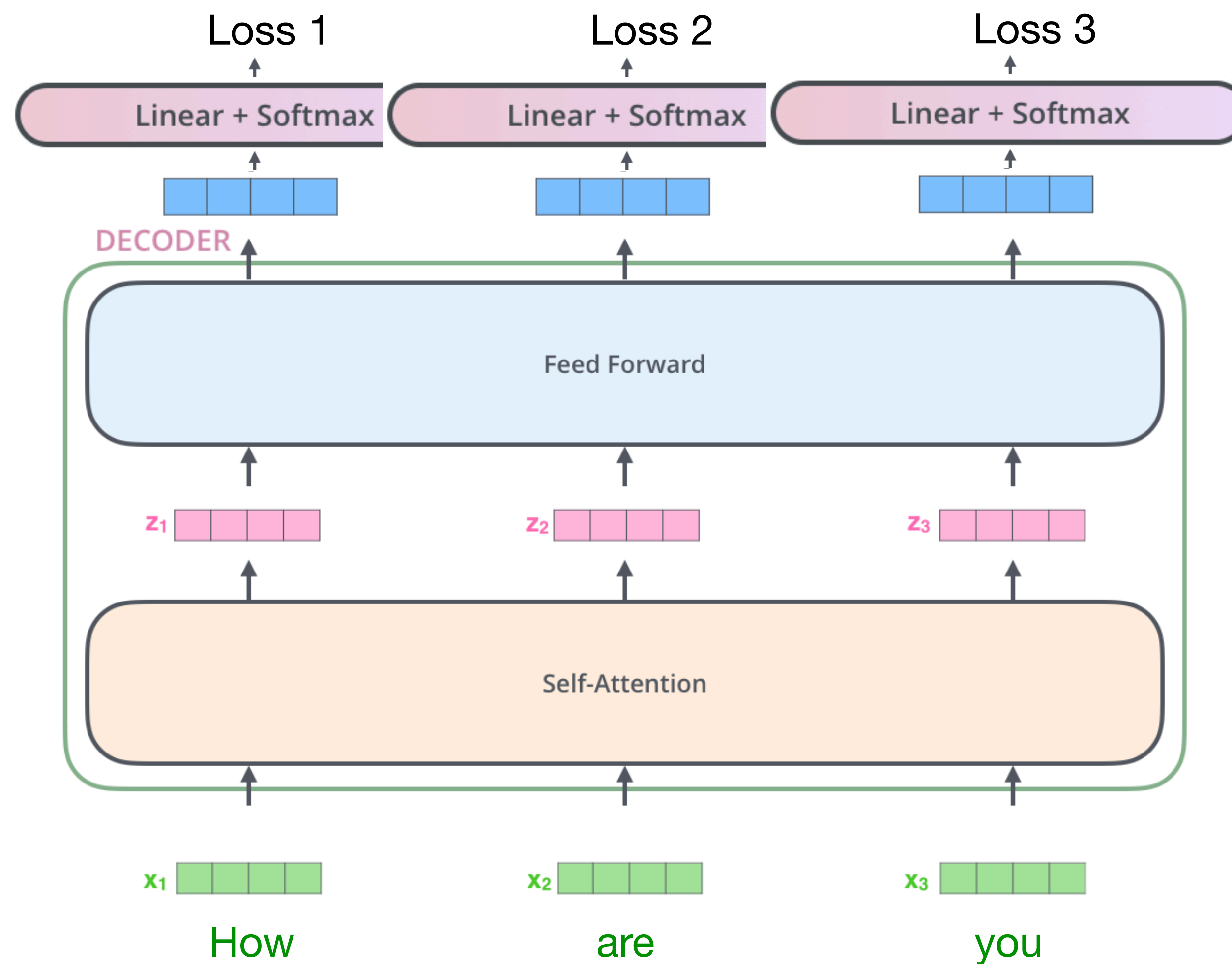


Encoder: одновременно обрабатываем

Decoder: последовательно

Encoder vs Decoder

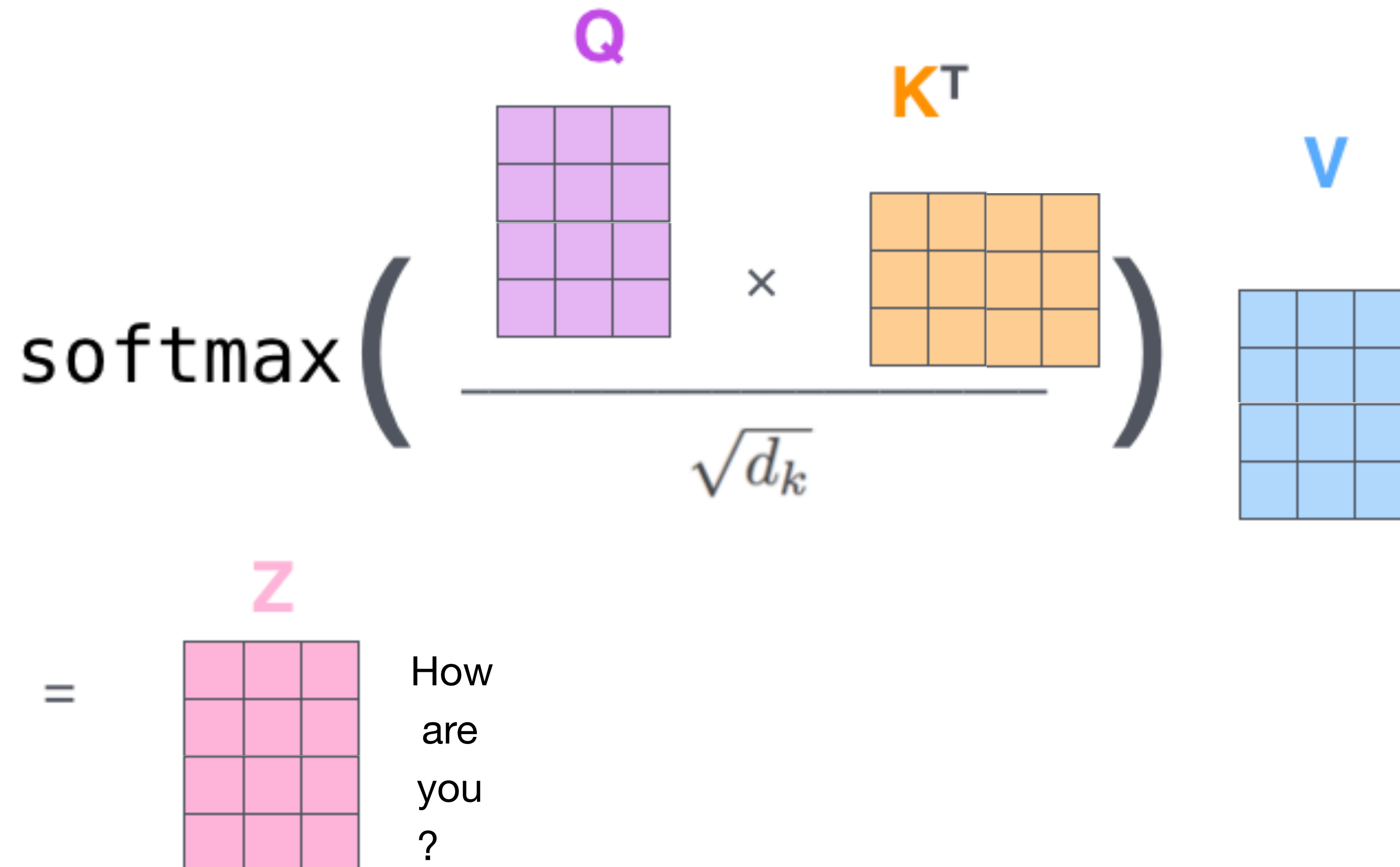
Train



Decoder: можно посчитать одновременно

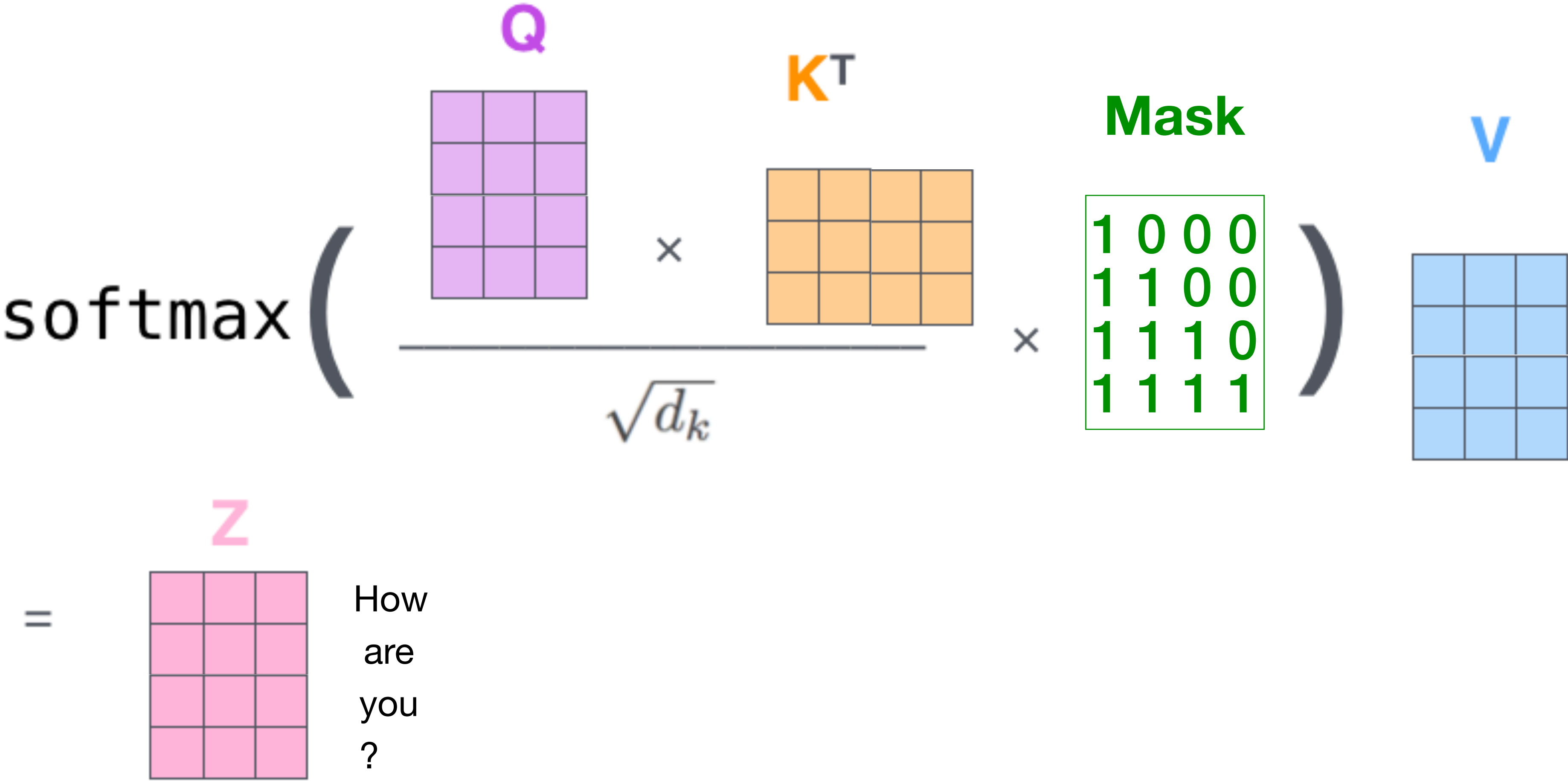
Encoder vs Decoder

Encoder self-attention: $attention = softmax(\frac{QK^T}{d})V$



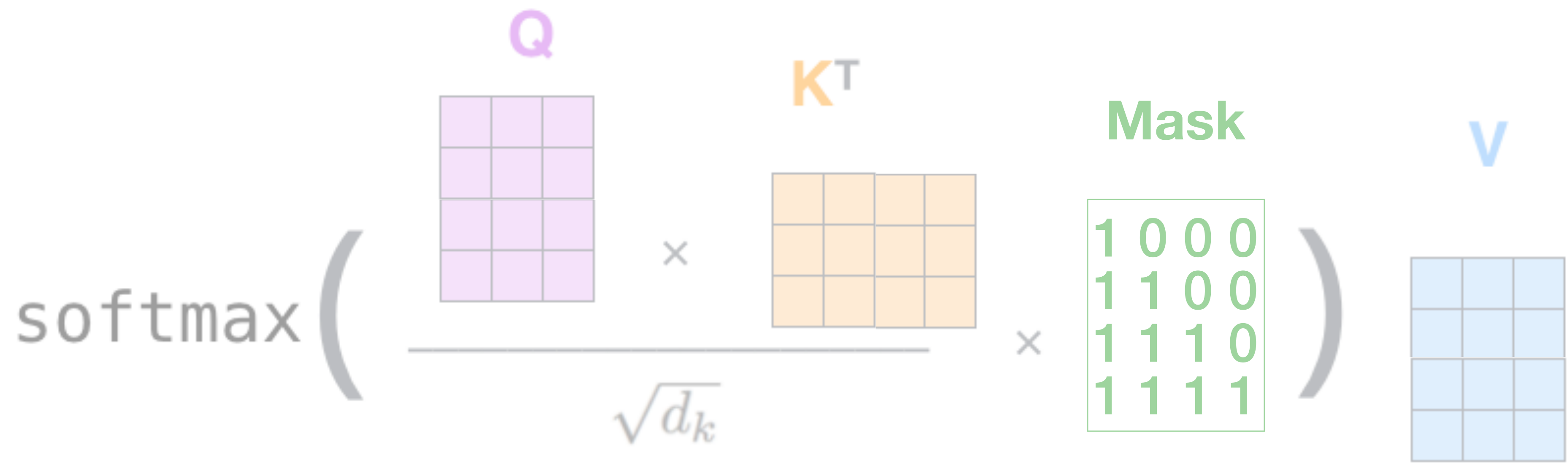
Encoder vs Decoder

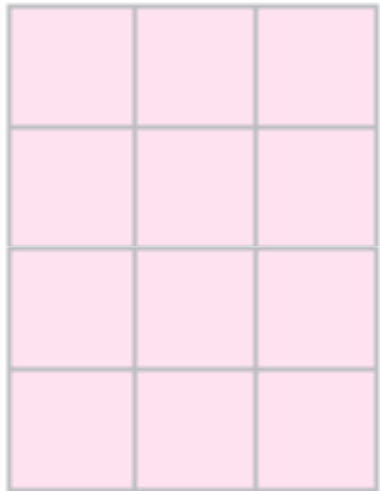
Decoder (masked) self-attention: $attention = softmax(\frac{QK^T + Mask}{d})V$

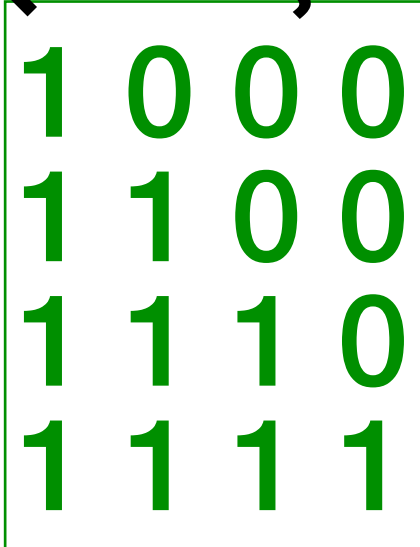


Encoder vs Decoder

Decoder (masked) self-attention: $attention = softmax(\frac{QK^T + Mask}{d})V$



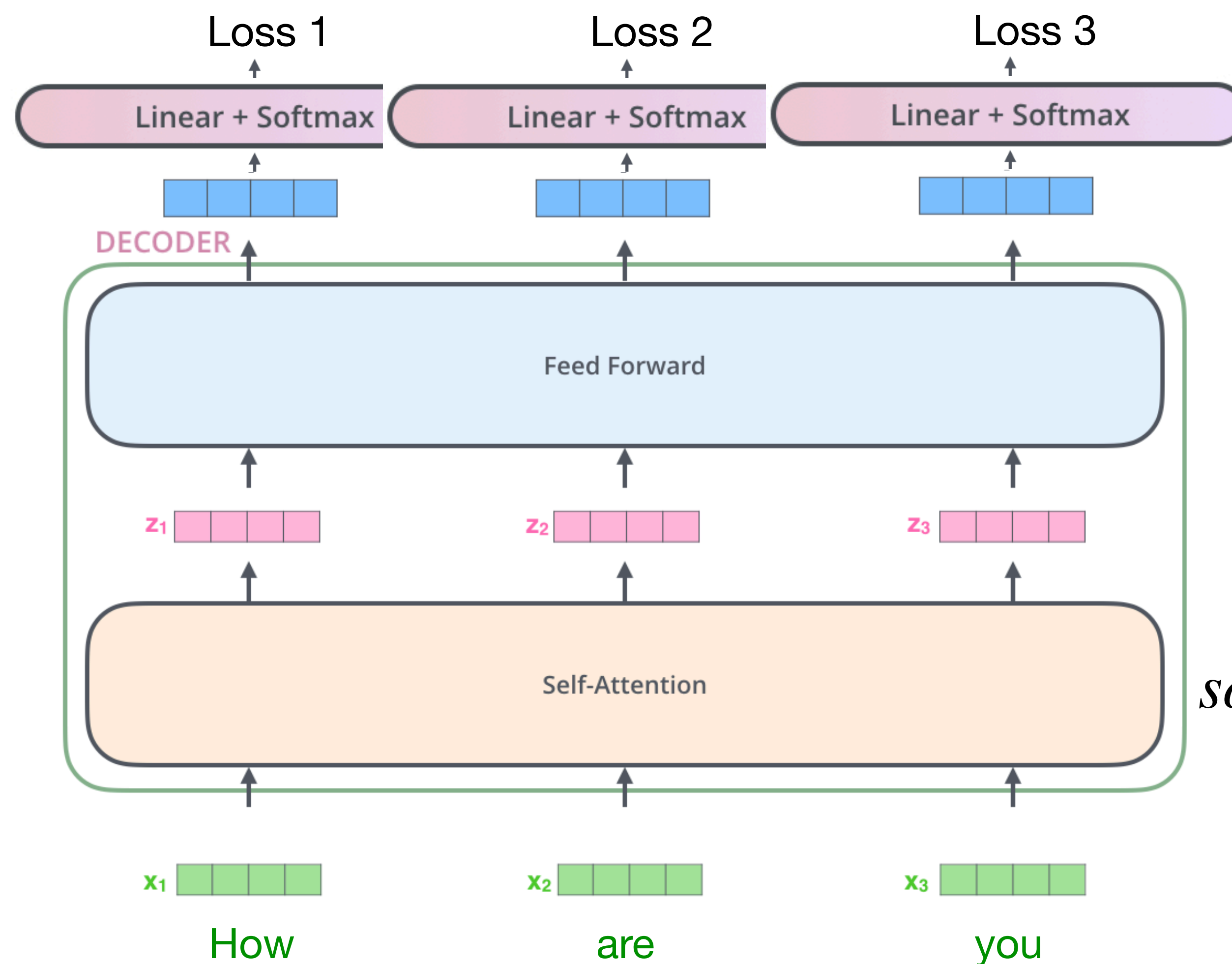
=  How are you ?

How are you ? 

Masked Attention
не будет смотреть
вперед

Encoder vs Decoder

Train



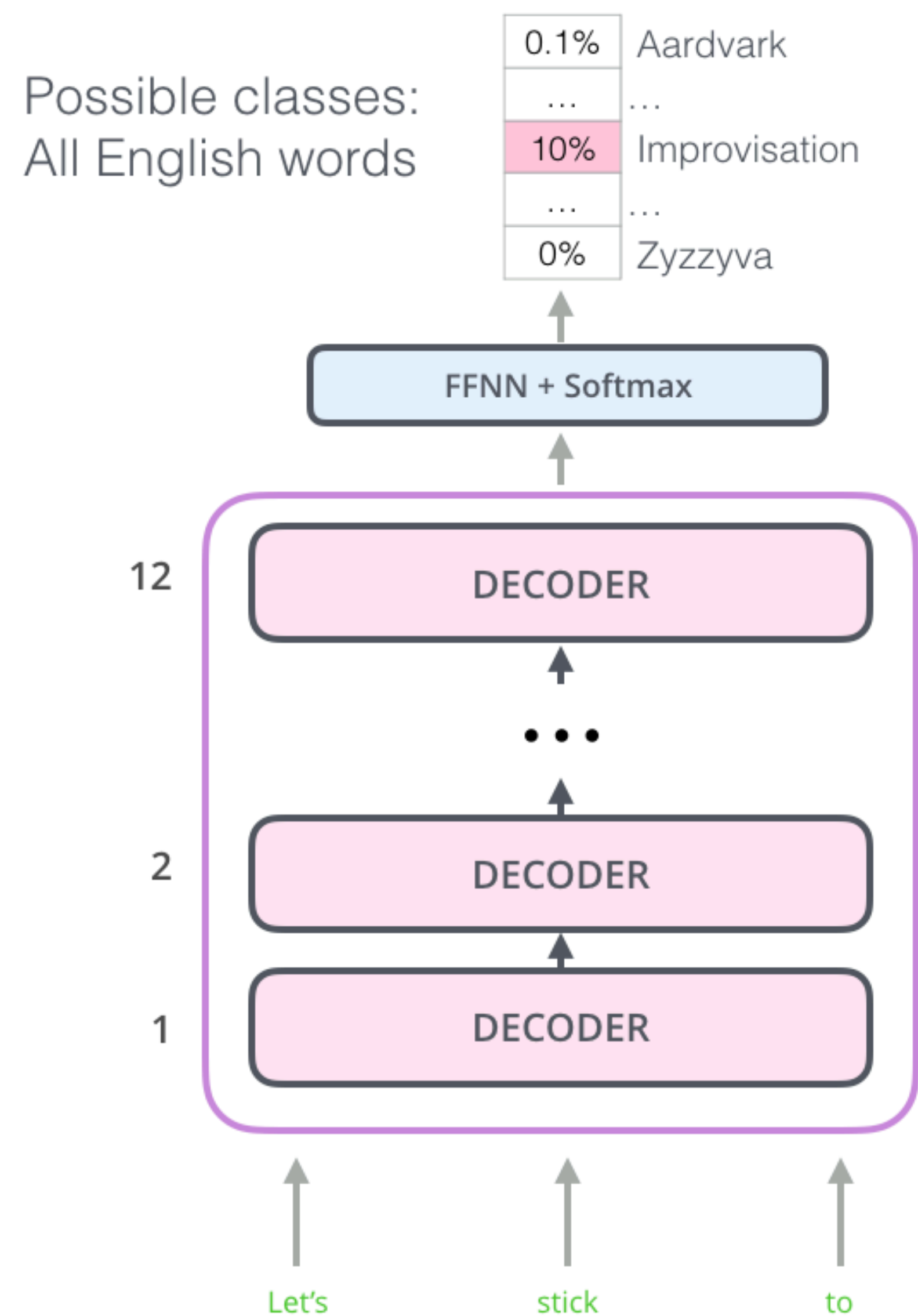
$$\text{softmax}\left(\frac{QK^T + \text{Mask}}{d}\right)V$$

Decoder: можно посчитать одновременно

GPT

Generative Pre-Training

Архитектура: Transformer Decoder



GPT

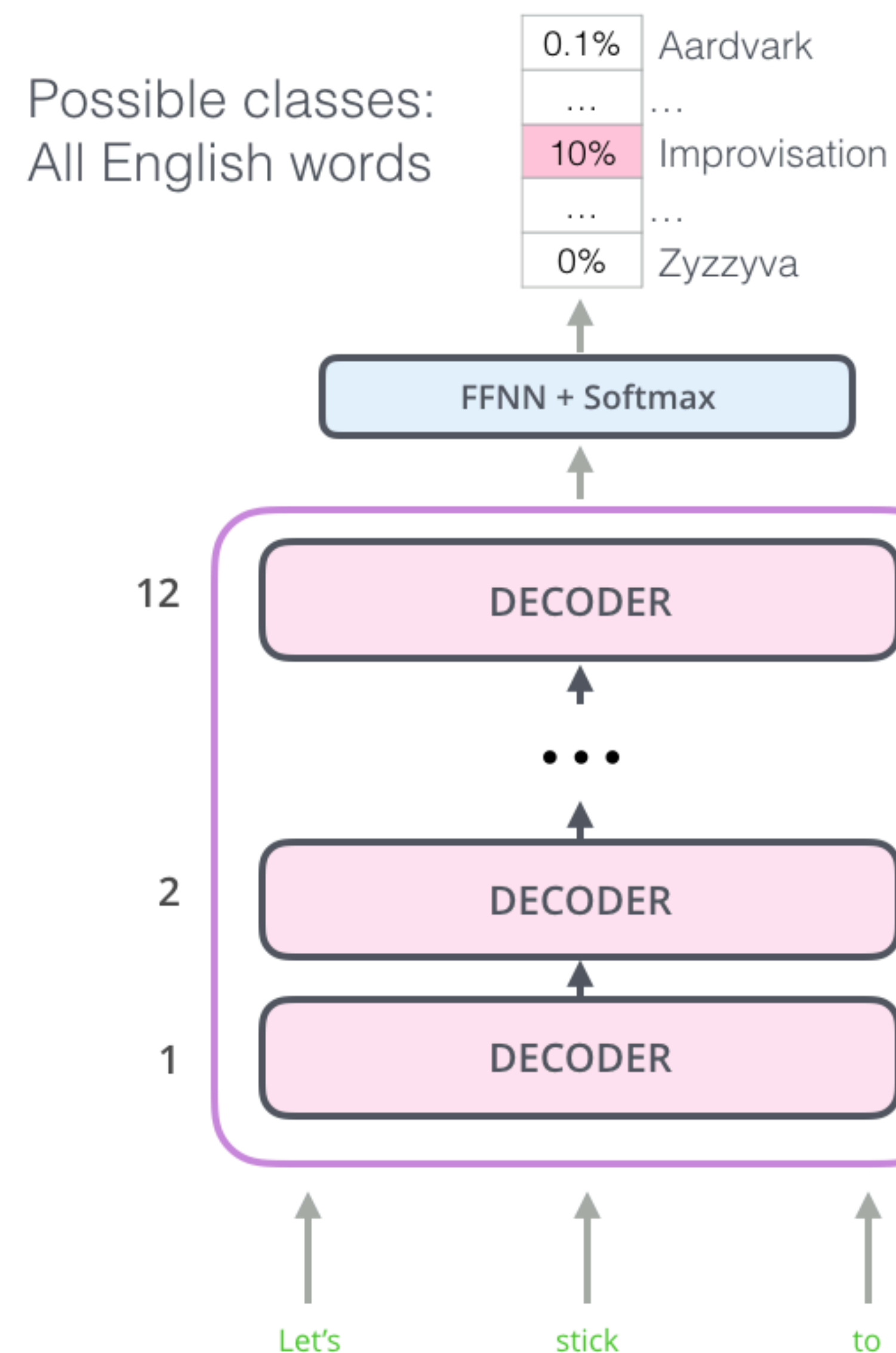
Generative Pre-Training

Архитектура: Transformer Decoder

Данные: тексты книг (BookCorpus)

+ Разнообразие

+ Большой объем



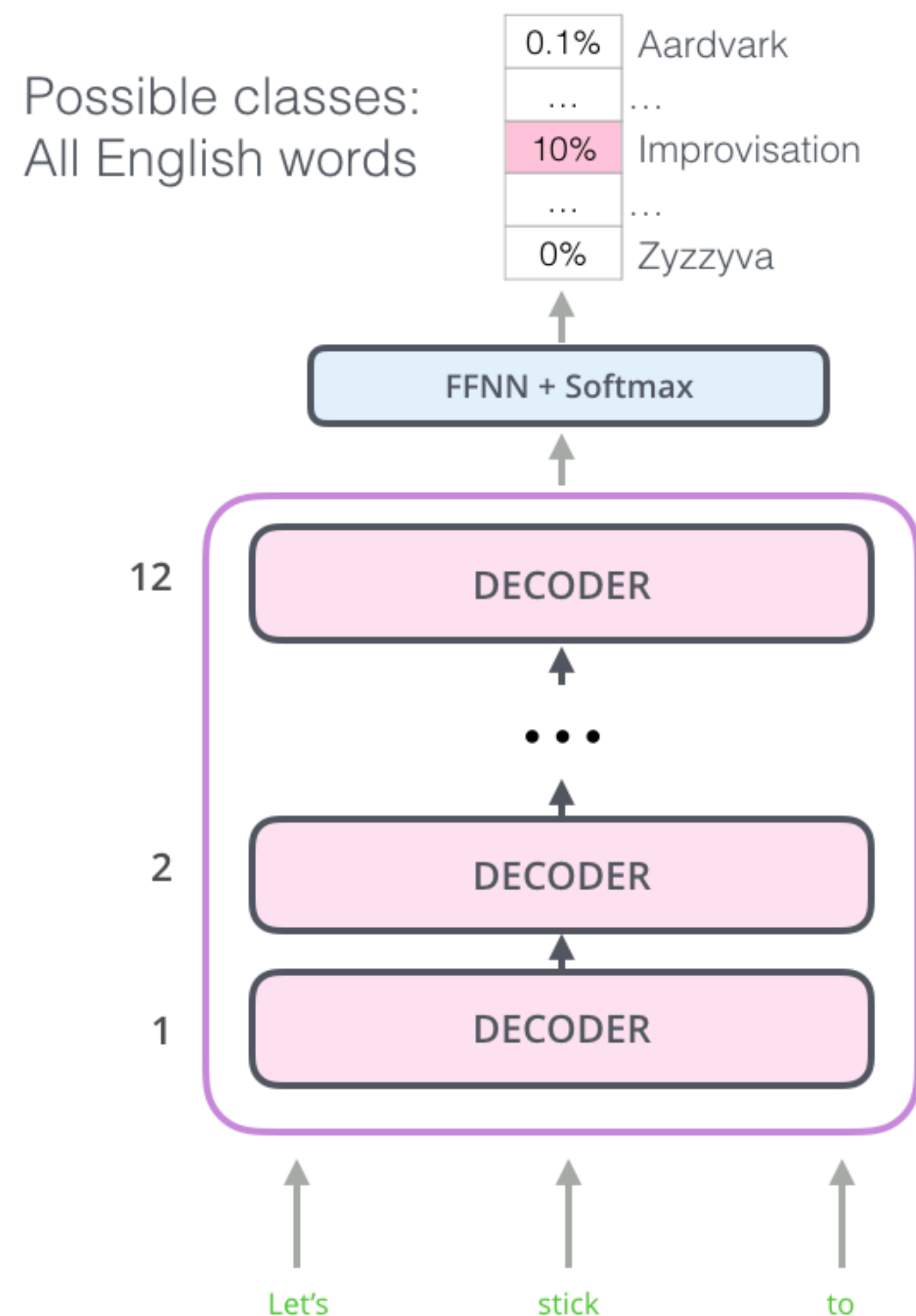
GPT

Generative Pre-Training

Архитектура: Transformer Decoder

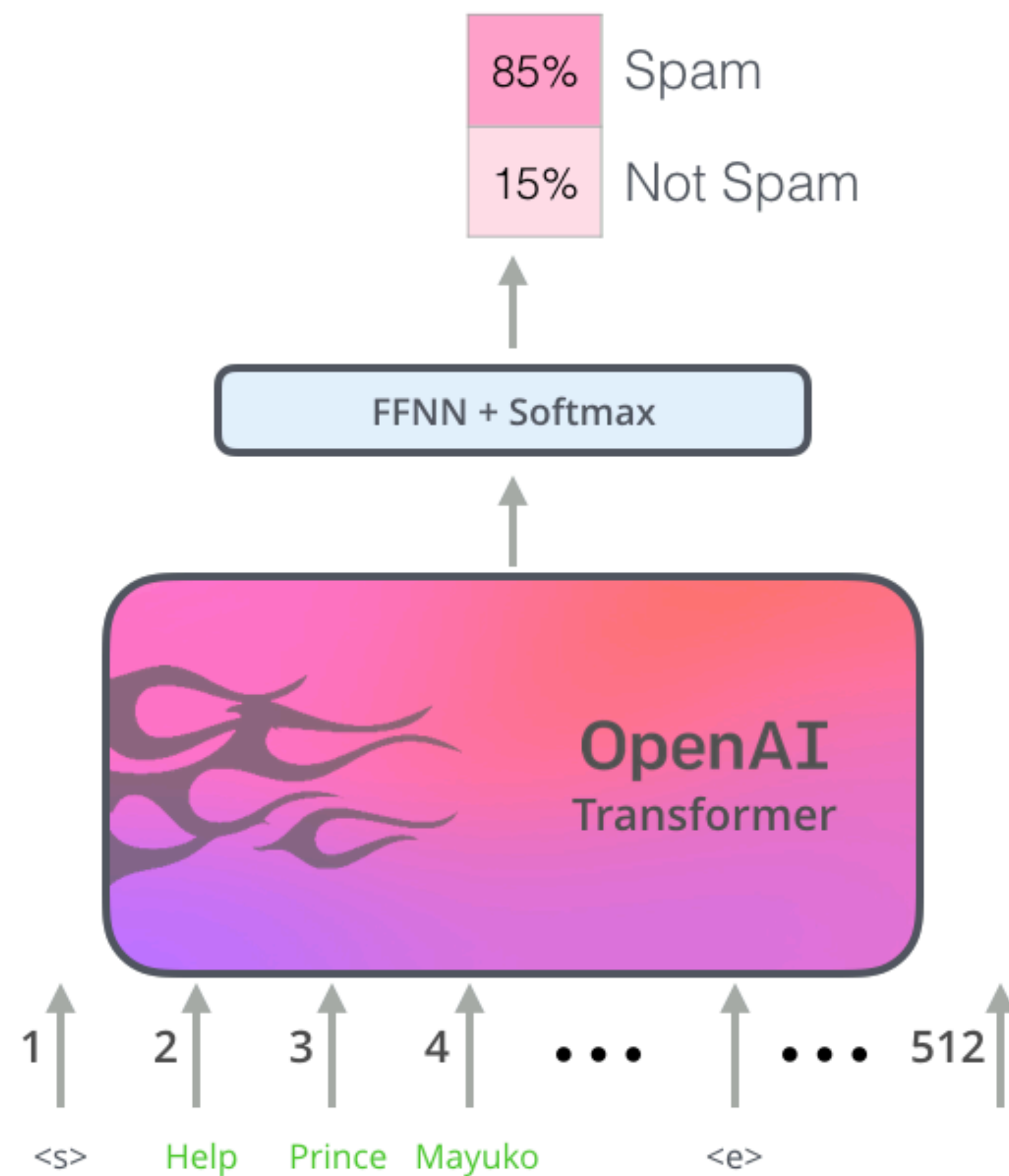
Данные: тексты книг (BookCorpus)

Задача для обучения: Language Modeling



GPT

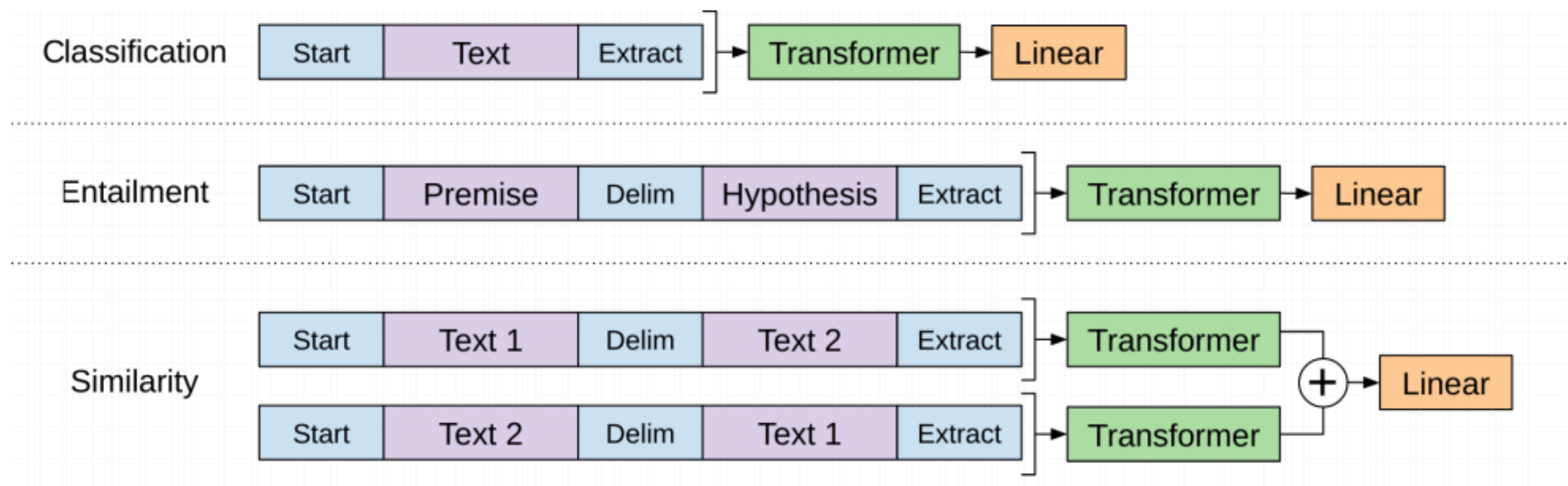
Как использовать?



GPT

Как использовать?

Для разных задач - разный формат входа



GPT-2

x10 параметров

x10 данных

GPT-2

Как использовать? **zero-shot task transfer**

Нет дообучения на новую задачу (fine-tuning)

Форматируем input, чтобы была понятна задача

GPT-2

Форматируем input, чтобы была понятна задача

Пример: задача суммаризации

Article: Amina Ali Qassim is sitting with her youngest grandchild on her lap, wiping away tears with her headscarf. Only a few months old, this is the baby girl whose ears she desperately tried to cover the night the aerial bombardment started. She lay awake, she says, in a village mosque on the Yemeni island of Birim, counting explosions as the baby cried.

It could have been worse though. They could have still been in their house when the first missile landed.

"Our neighbor shouted to my husband 'you have to leave, they're coming.' And we just ran. As soon as we left the house, the first missile fell right by it and then a second on it. It burned everything to the ground," Qassim tells us ...

TL;DR:

input

GPT-2

Форматируем input, чтобы была понятна задача

Пример: задача суммаризации

Article: Amina Ali Qassim is sitting with her youngest grandchild on her lap, wiping away tears with her headscarf. Only a few months old, this is the baby girl whose ears she desperately tried to cover the night the aerial bombardment started. She lay awake, she says, in a village mosque on the Yemeni island of Birim, counting explosions as the baby cried.

It could have been worse though. They could have still been in their house when the first missile landed.

"Our neighbor shouted to my husband 'you have to leave, they're coming.' And we just ran. As soon as we left the house, the first missile fell right by it and then a second on it. It burned everything to the ground," Qassim tells us ...

TL;DR: Yemen is in the middle of a civil war. Saudi Arabia is leading the coalition bombing campaign. It's been bombing Yemen for more than two months now.



GPT-2 prediction

Input

GPT-2

Language Modeling (генерация текста)

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.



GPT-3

x100 параметров (по сравнению с GPT-2)

x5 данных (по сравнению с GPT-2)

* нет в свободном доступе

GPT-3

x100 параметров (по сравнению с GPT-2)

x5 данных (по сравнению с GPT-2)

Как использовать? **zero-shot/one-shot/few-shot settings**

- показываем 0/1/несколько примеров (prompts)


GPT-3

х100 параметров (по сравнению с GPT-2)

х5 данных (по сравнению с GPT-2)

Как использовать? **zero-shot/one-shot/few-shot settings**

- показываем 0/1/несколько примеров (prompts)

English sentence 1	=	France sentence 1	Input
English sentence 2	=	France sentence 2	
English sentence 3	=		
			
GPT-3 prediction			

GPT-3

Prompt:

The following are ideas for startups, apps, marketplaces, platforms:

###

A startup that helps content creators, such as photographers and videographers, monetize their work.

###

An online forum that helps people find roommates

###

GPT-3:

A community hub for entrepreneurs to share ideas and ask questions of experienced entrepreneurs

###

A website that helps people discover new interests and hobbies.

###

A company that helps brands advertise their products to potential buyers on Instagram.

###

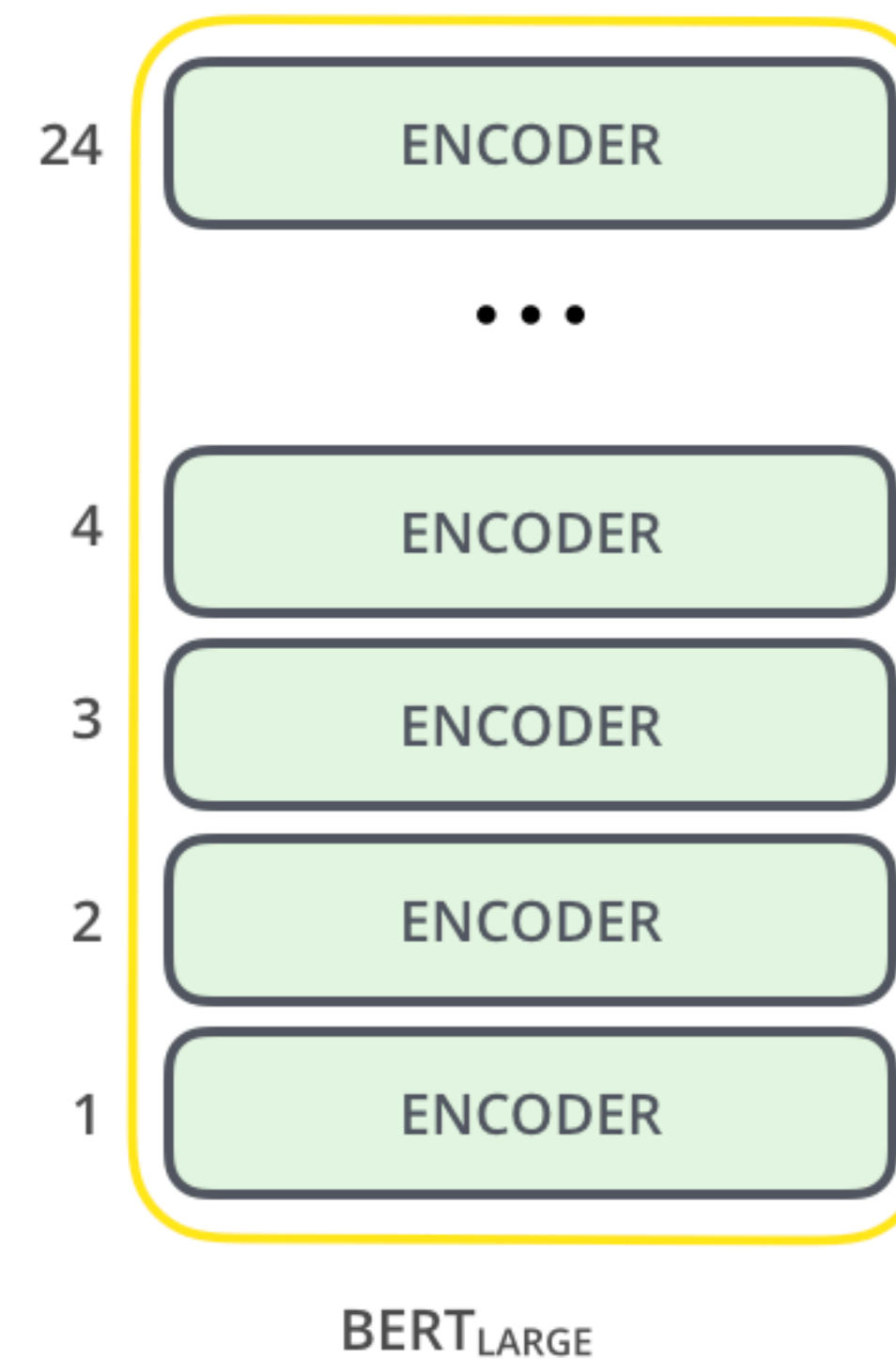
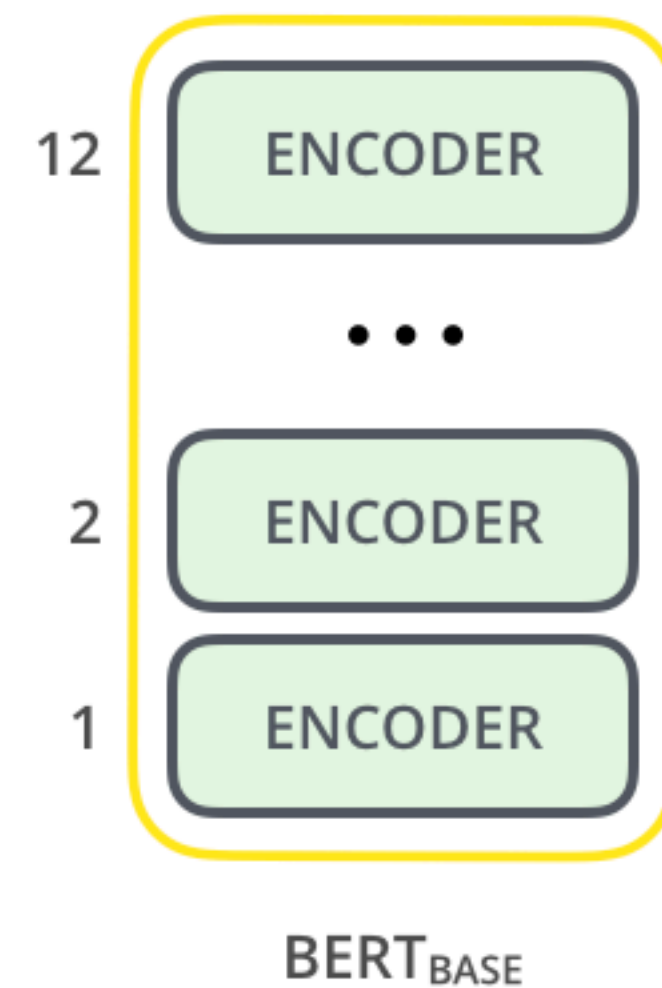


BERT

BERT

Bidirectional Encoder Representations from Transformers

Архитектура: Transformer Encoder



BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

Архитектура: Transformer Encoder

Данные: Wikipedia и тексты книг (BookCorpus)

BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

Архитектура: Transformer Encoder

Данные: Wikipedia и тексты книг (BookCorpus)

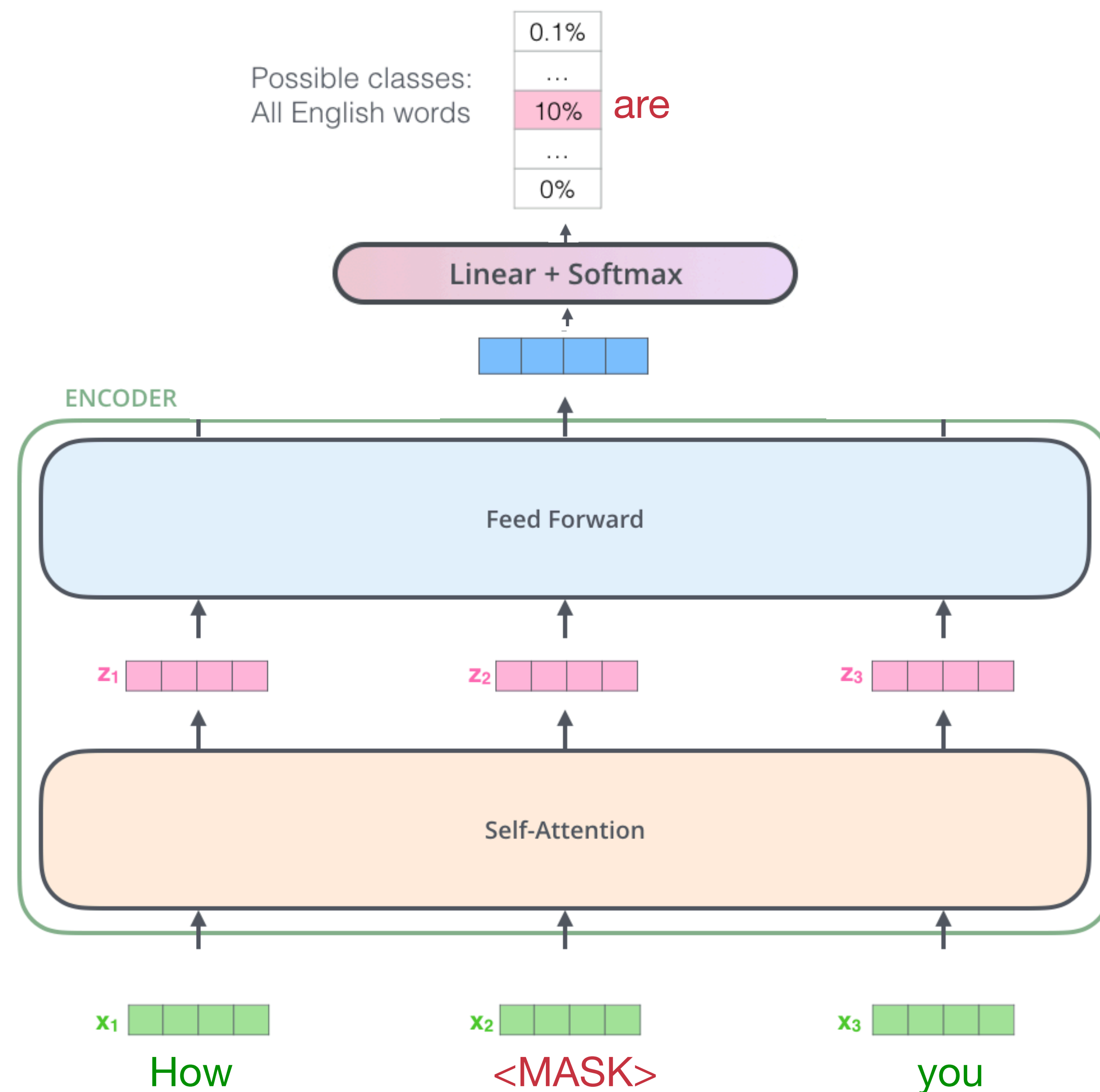
Задачи для обучения: Masked LM и Next Sentence Prediction

BERT

Masked Language Model

Случайно выбираем 15% позиций и заменяем на <MASK>

Задача: предсказать исходный токен



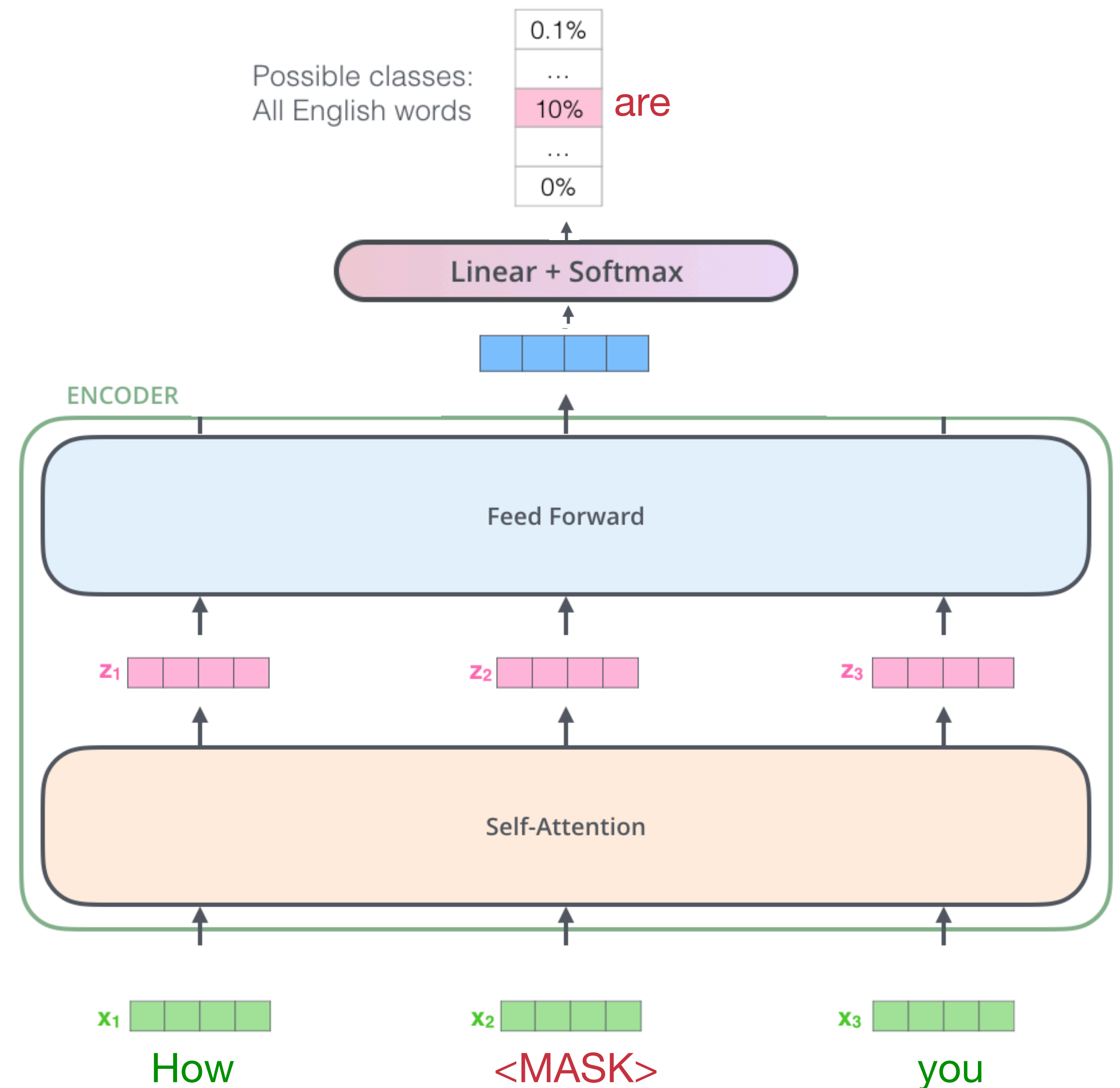
BERT

Masked Language Model

Случайно выбираем 15% позиций:

- 80% заменяем на <MASK>
- 10% заменяем на случайный токен
- 10% оставляем

Задача: предсказать исходный токен



BERT

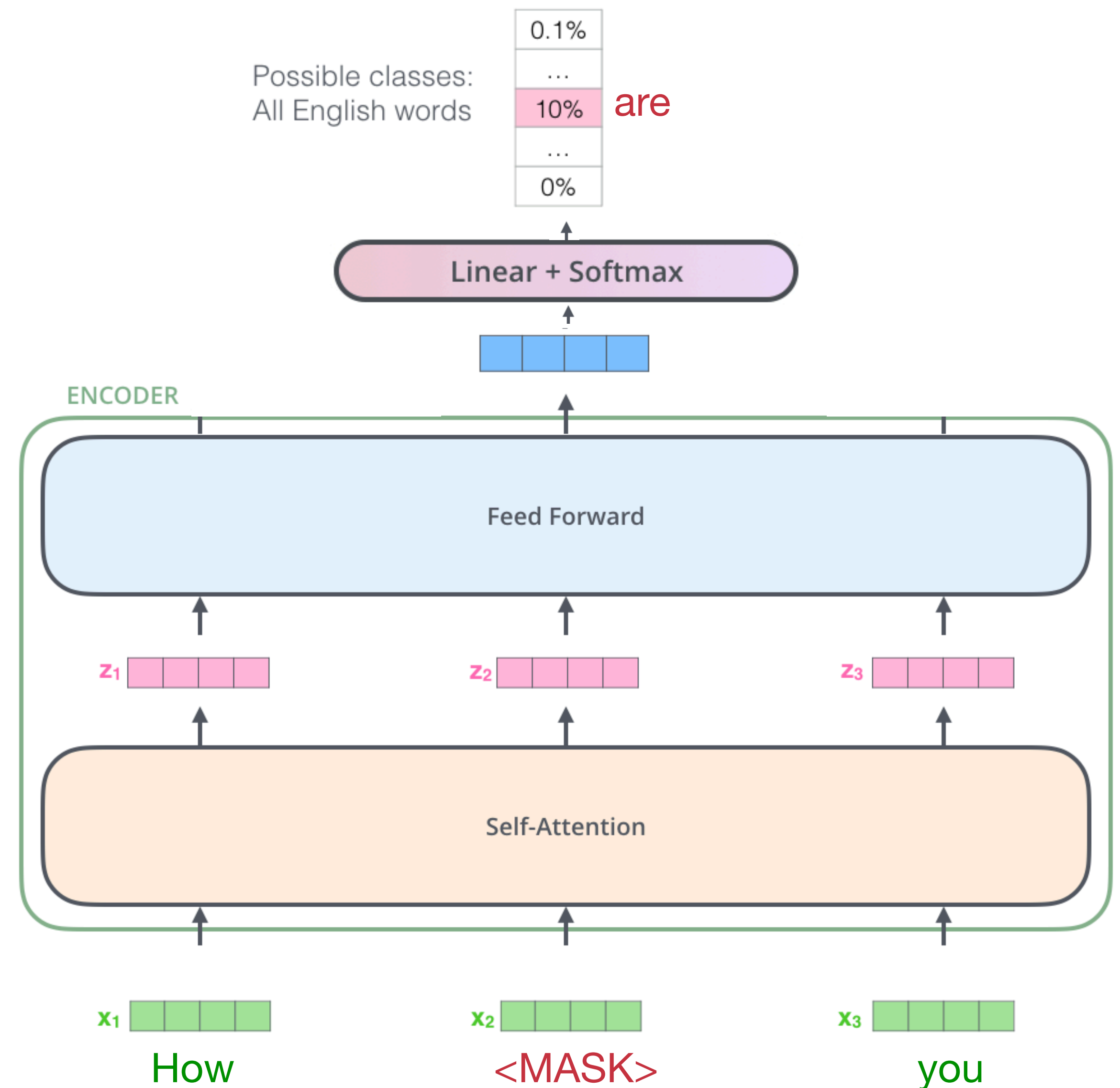
Masked Language Model

Случайно выбираем 15% позиций:

- 80% заменяем на <MASK>
- 10% заменяем на случайный токен
- 10% оставляем

Задача: предсказать исходный токен

Обученные эмбеддинги учитывают
контекст слева и справа



BERT

Next Sentence Prediction

Для некоторых задач нужно понимать взаимоотношения между двумя предложениями:

- Similarity
- Entailment
- Question Answering
- ...

BERT

Next Sentence Prediction

Вход: 2 предложения (A и B)

50% - B следует за A в тексте

50% - B выбрано случайно

Формат входа:

[CLS] the man [MASK] to the store [SEP] penguin [MASK] are flightless birds [SEP]

└────────────────────────────────┘ └────────────────────────────────┘

Sentence A Sentence B

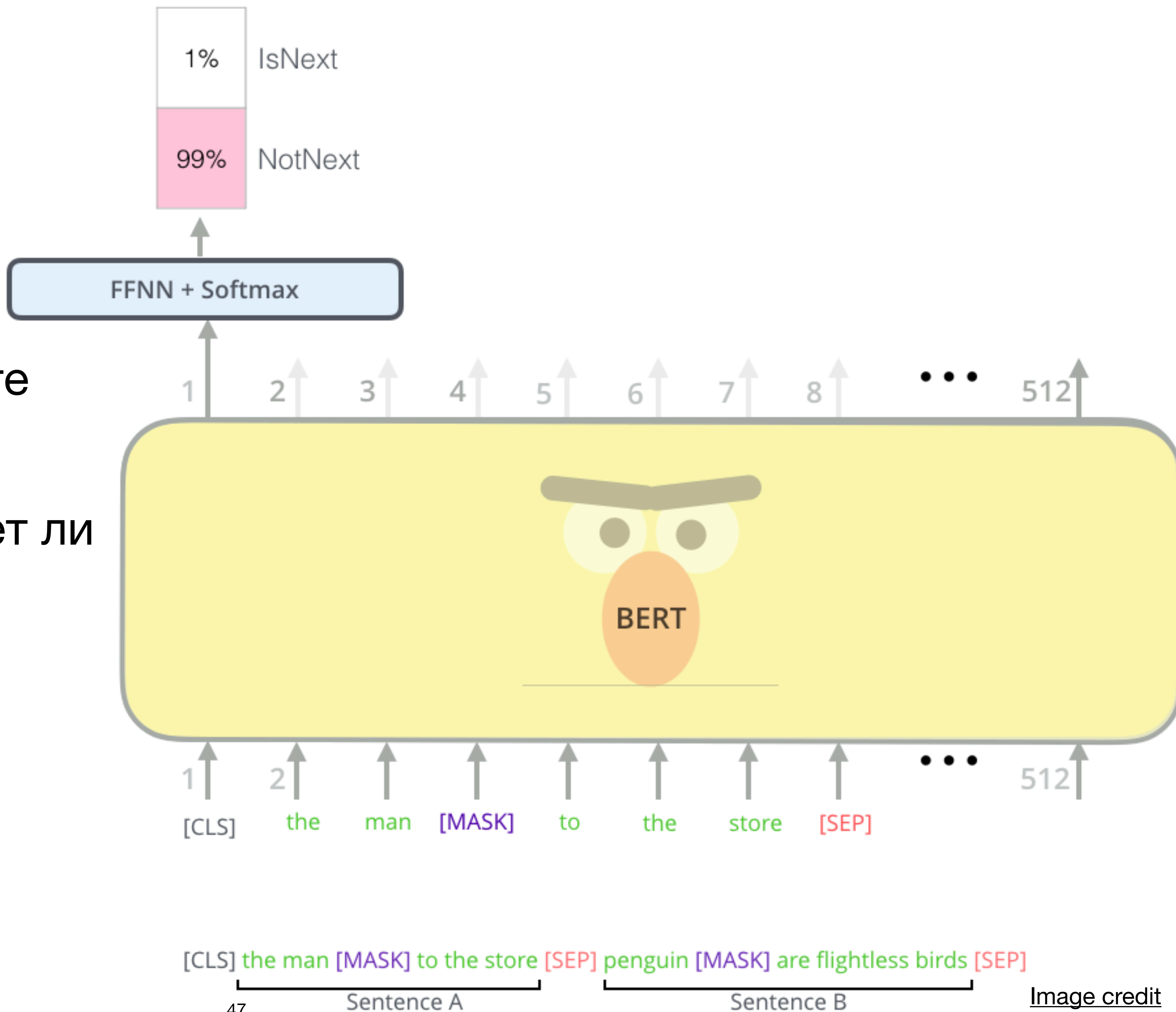
BERT

Next Sentence Prediction

Вход: 2 предложения (A и B)

50% - B следует за A в тексте
50% - B выбрано случайно

Задача: предсказать, следует ли
B за A



BERT

Next Sentence Prediction

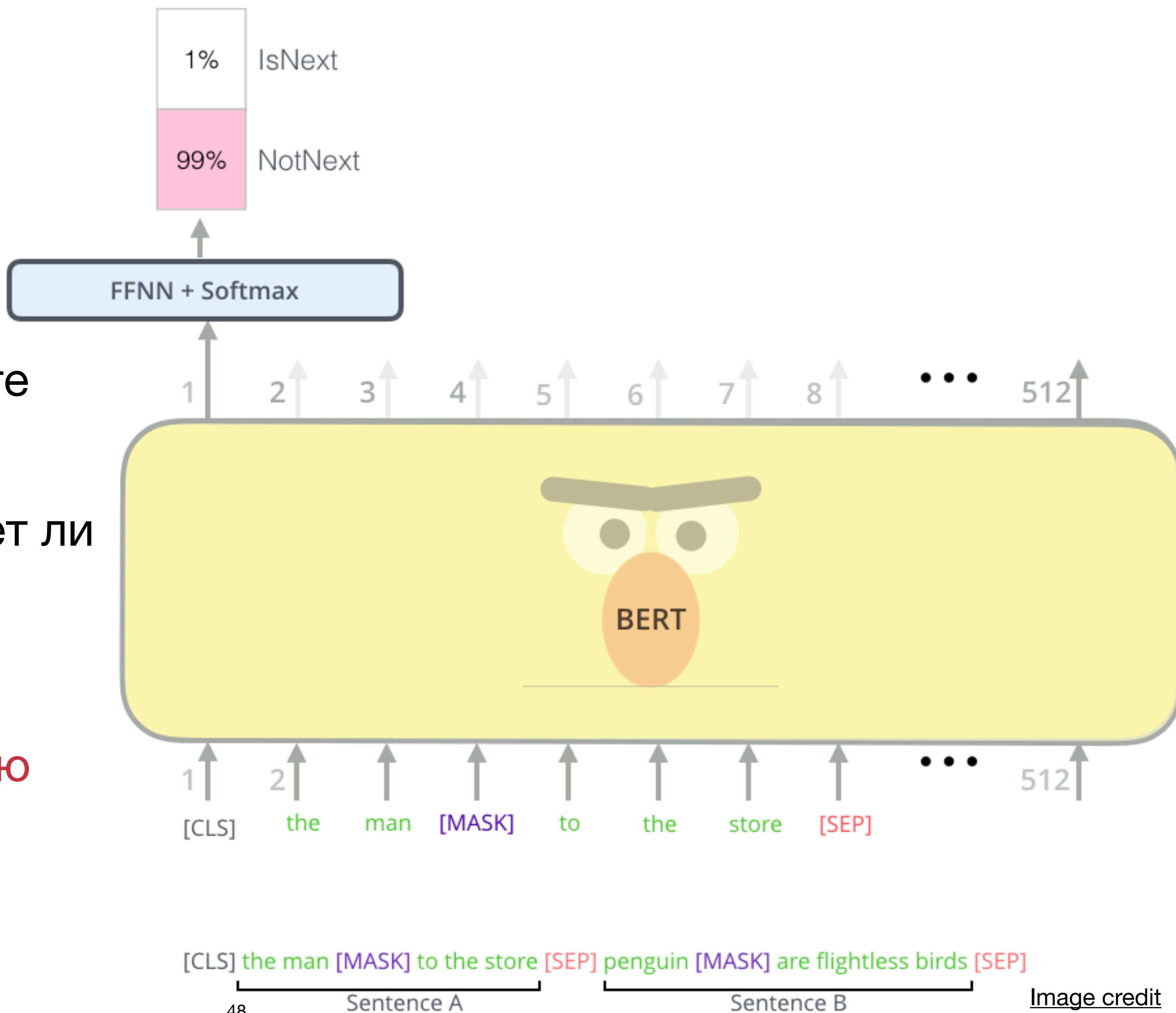
Вход: 2 предложения (A и B)

50% - B следует за A в тексте
50% - B выбрано случайно

Задача: предсказать, следует ли
B за A

<CLS> - выучивает
агрегированную информацию

<SEP> - разделитель



BERT

Как использовать?

- Linear+Softmax поверх <CLS> - для задач классификации предложения (или двух)
- Linear+Softmax поверх всех выходов - для задач классификации токенов
- Выходы BERT - как вход в другие модели (task-specific)
- ...

RoBERTa

Robustly Optimized **BERT** Pretraining Approach

Улучшенная версия BERT:

- x10 данных
- x10 вычислительных ресурсов (дольше обучение, больше batch size)
- Не использовали задачу Next Sentence Prediction

Улучшение в 2-20% (в зависимости от задачи)

BART and T5

BART

Bidirectional and **A**uto-**R**egressive **T**ransformers

Идея: соединить преимущества BERT и GPT

Архитектура: Transformer Encoder-Decoder

Данные: как в RoBERTa

BART

Bidirectional and Auto-Regressive Transformers

Идея: соединить преимущества BERT и GPT

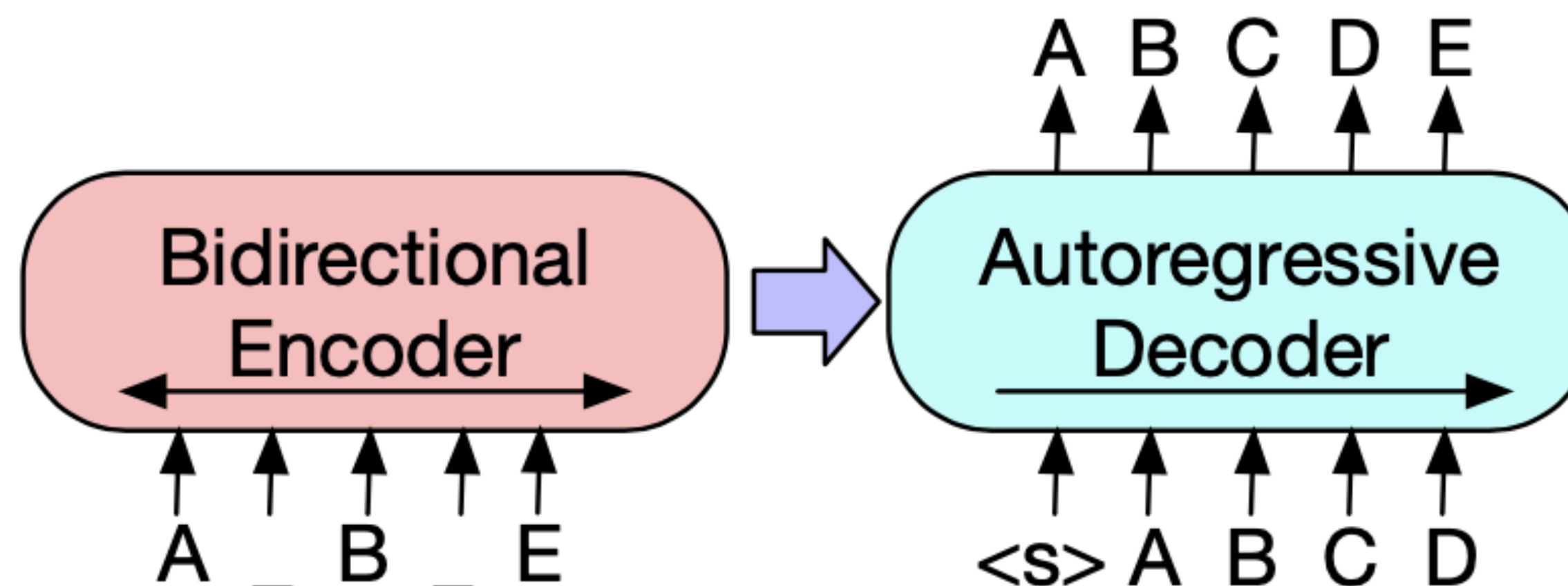
Архитектура: Transformer Encoder-Decoder

Данные: как в RoBERTa

Задача для обучения: восстановить последовательность

Последовательность: ABCDE

- Маскируем C, D
- Шум: лишняя маска перед B



T5

Text-to-Text Transfer Transformer

Идея: соединить преимущества BERT и GPT

Архитектура: Transformer Encoder-Decoder

Данные: x5 от данных RoBERTa

Задача для обучения: восстановить последовательность

T5

Text-to-Text Transfer Transformer

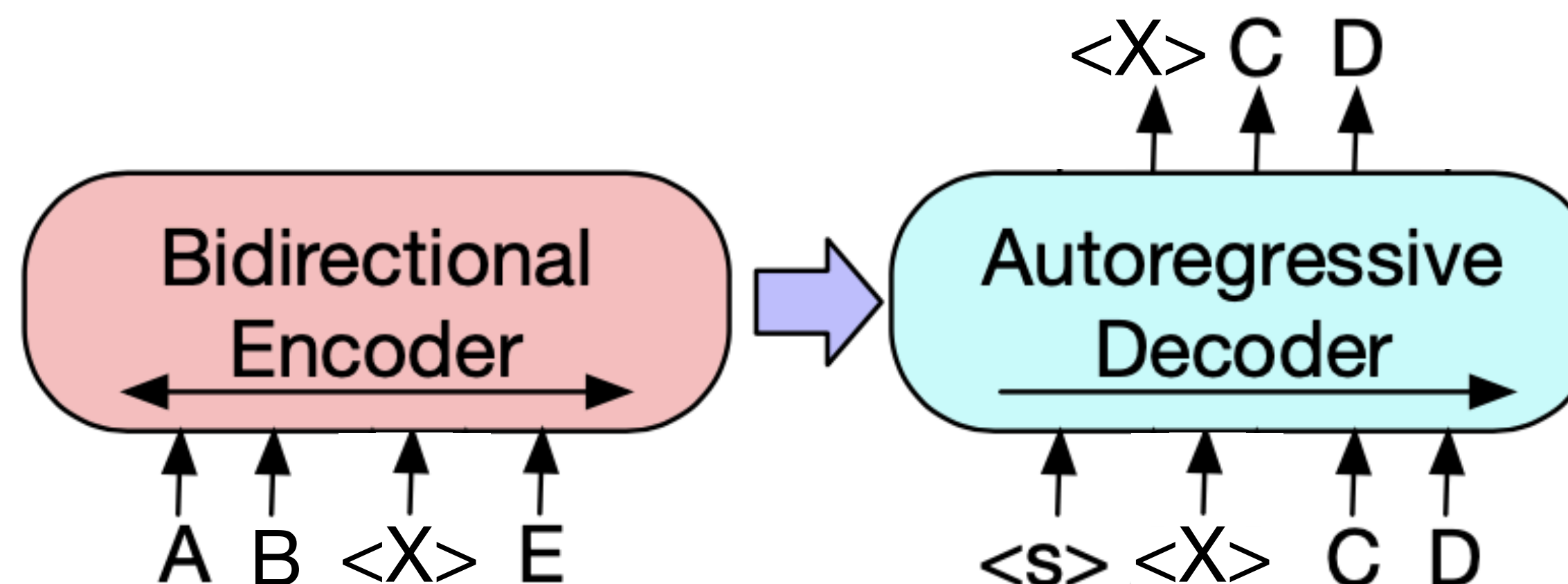
Идея: соединить преимущества BERT и GPT

Архитектура: Transformer Encoder-Decoder

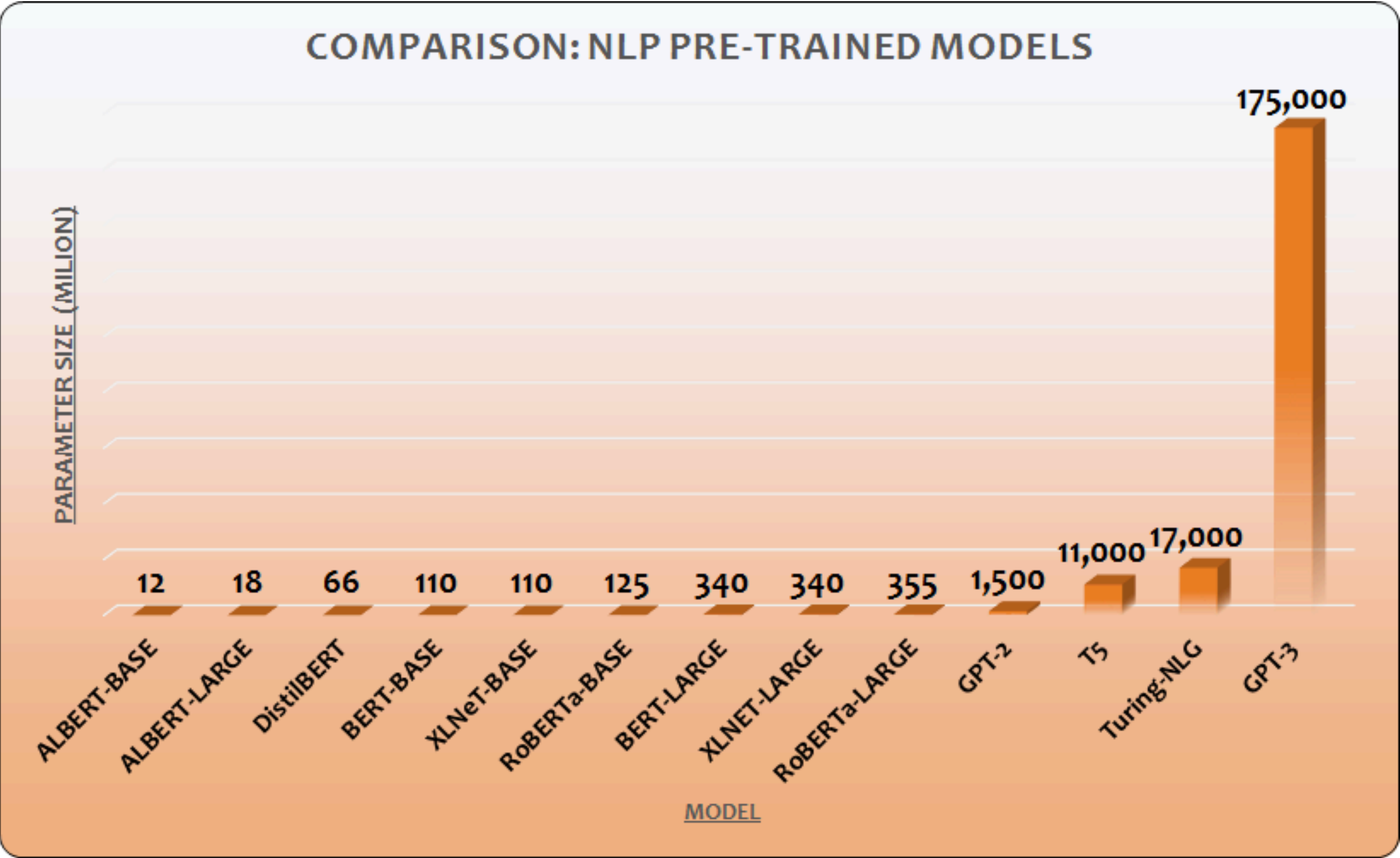
Данные: x5 от данных RoBERTa

Задача для обучения: ВОССТАНОВИТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ

Вместо маски используем
специальные токены

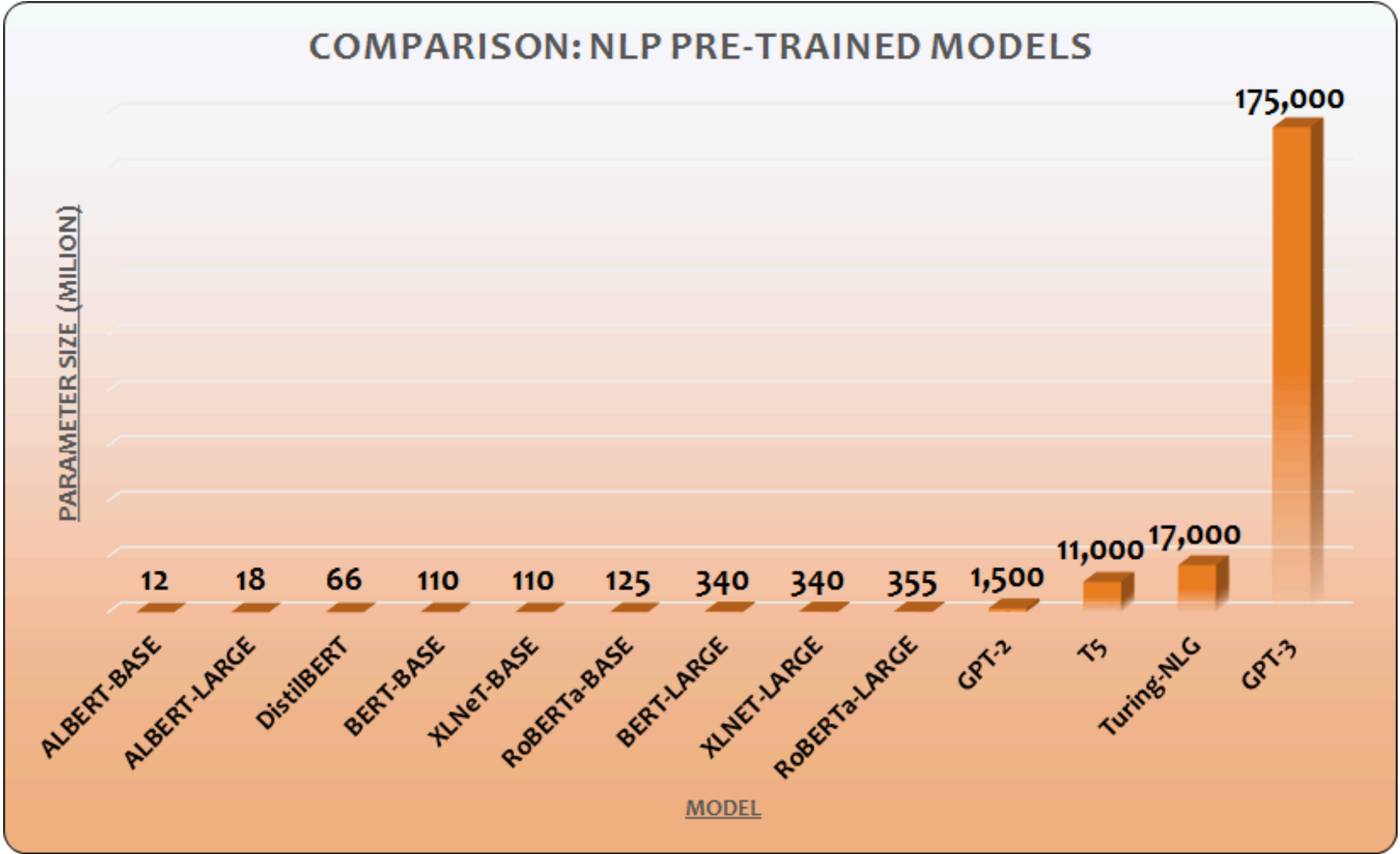


Сравнение



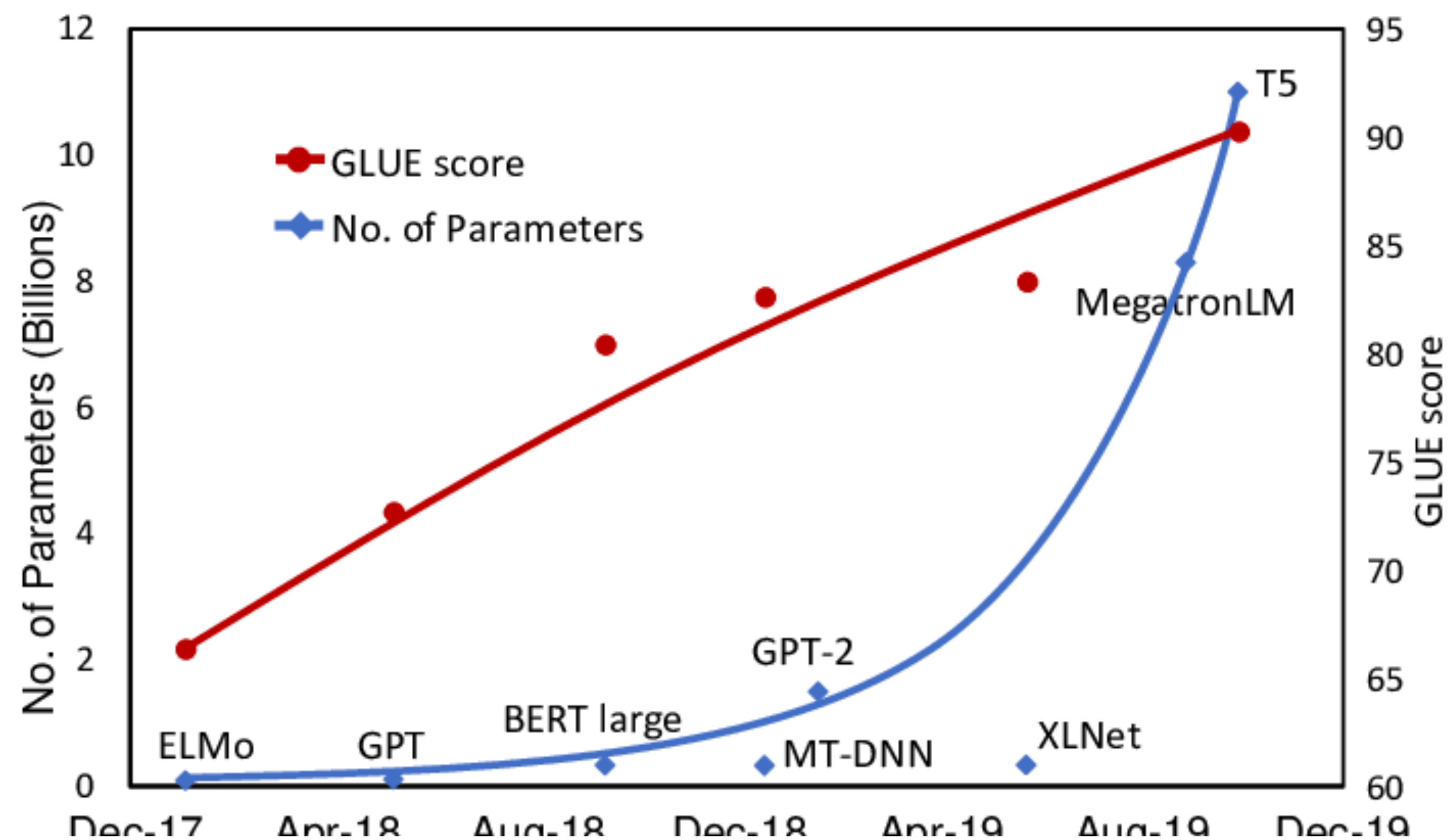
Сравнение

~ 2 недели назад: PaLM от Google - 540,000!



Сравнение

Leaderboards: GLUE, SuperGLUE



Как использовать?

Большой список доступных моделей и удобный интерфейс - библиотека HuggingFace Transformers 🤗

Как использовать?

Большой список доступных моделей и удобный интерфейс - библиотека HuggingFace Transformers 🤗

- GPT-2 - хорошая генерация текста
- BERT, RoBERTa - классификация, использование в качестве эмбедингов
- T5, BART - seq2seq задачи

Как использовать?

Большой список доступных моделей и удобный интерфейс - библиотека HuggingFace Transformers 🤗

Другие модели:

- Улучшения в качестве
- Увеличение эффективности
- Работа с длинными последовательностями
- Специфичность к задаче/языку

Как использовать?

Большой список доступных моделей и удобный интерфейс - библиотека HuggingFace Transformers 🤗

Важная часть модели - **токенизатор**

Основной подход: разделить слово на часто встречаемые последовательности СИМВОЛОВ

	word		vocab mapping
Common words	hat	→	hat
	learn	→	learn
Variations	taaaaasty	→	taa## aaa## sty
misspellings	laern	→	la## ern
novel items	Transformerify	→	Transformer## ify

Разные алгоритмы: BPE, WordPiece, SentencePiece, ...