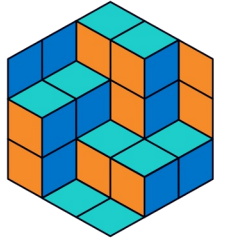


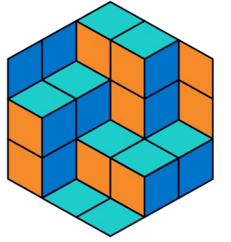
Курс: генерация рассказов

Часть 5: Greedy Search & Beam Search



Глава 1

Напоминание



Вот эти трансформеры из статьи “attention is all you need”

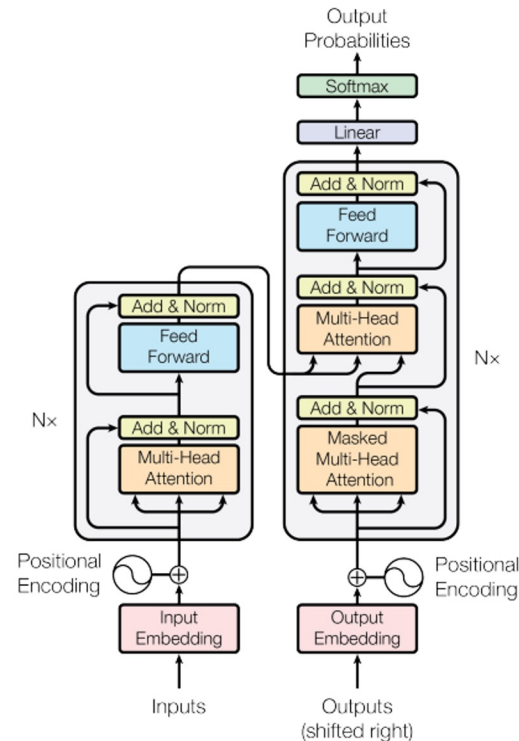
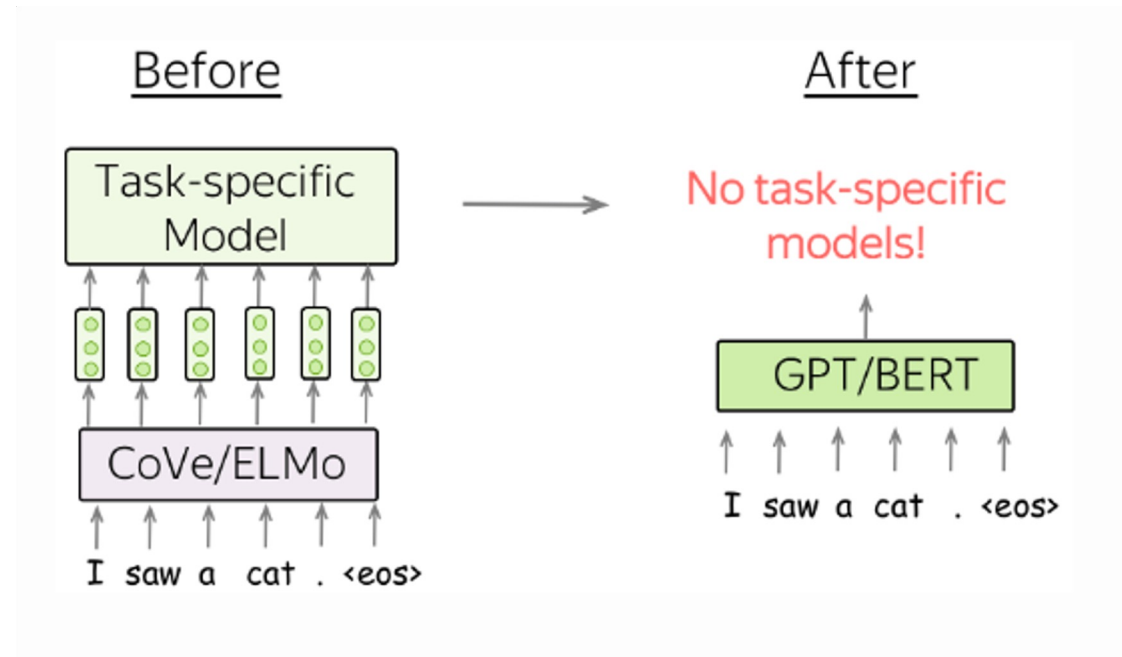
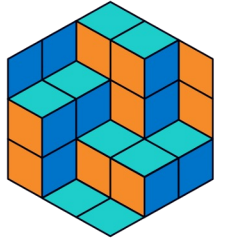
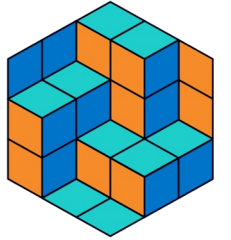


Figure 1: The Transformer - model architecture.





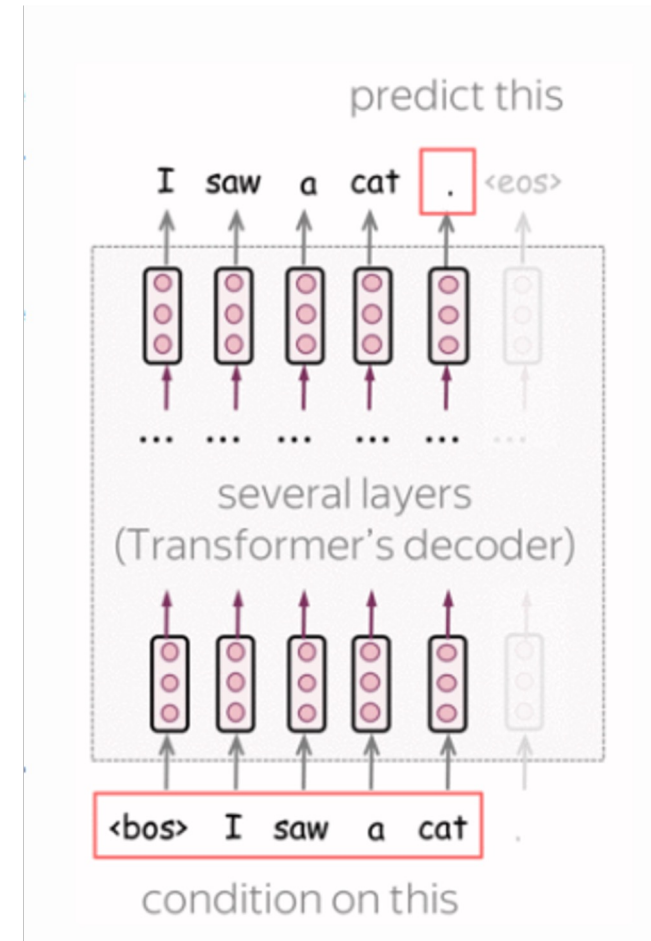
Знаменитая GPT

GPT по-сути это декодер трансформера.

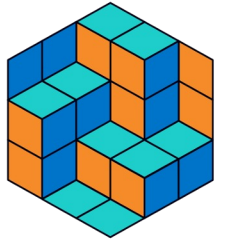
Он учится предсказывать токены в тексте по предыдущим.

Преимущество: не нужно размечать данные.

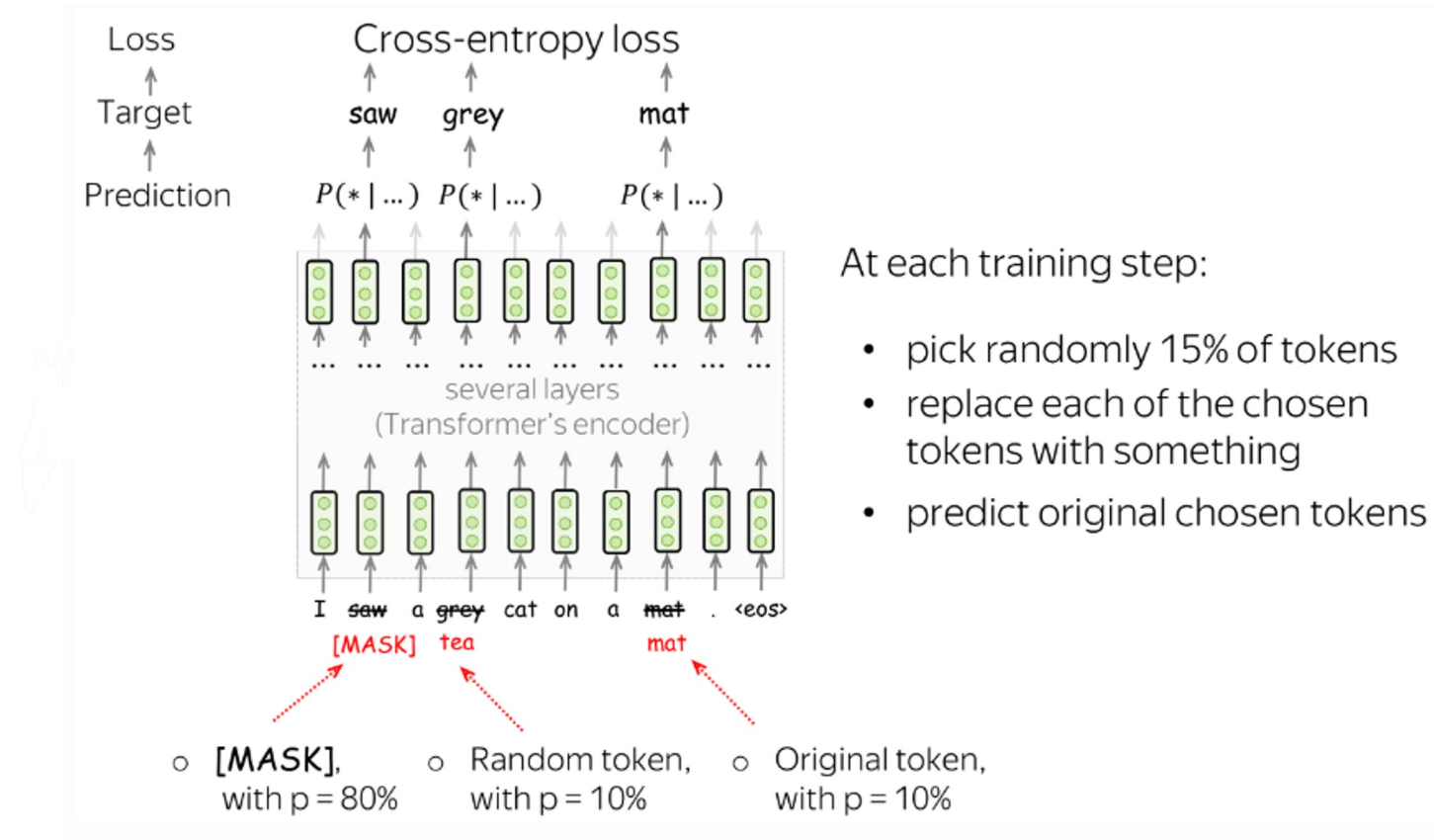
Недостатки: не умеет решать конкретные задачи.



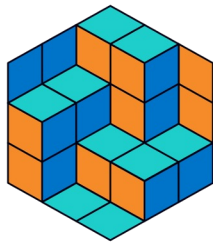
BERT: Bidirectional Encoder Representations from Transformers



BERT по сути это энкодер трансформера.

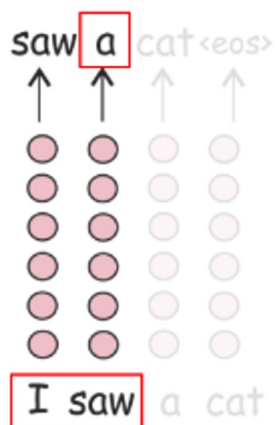


BERT



Language Modeling

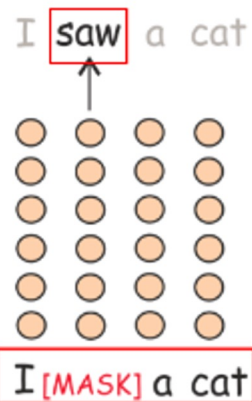
- Target: next token
- Prediction: $P(* | \text{I saw})$



left-to-right, does
not see future

Masked Language Modeling

- Target: current token (the true one)
- Prediction: $P(* | \text{I [MASK] a cat})$

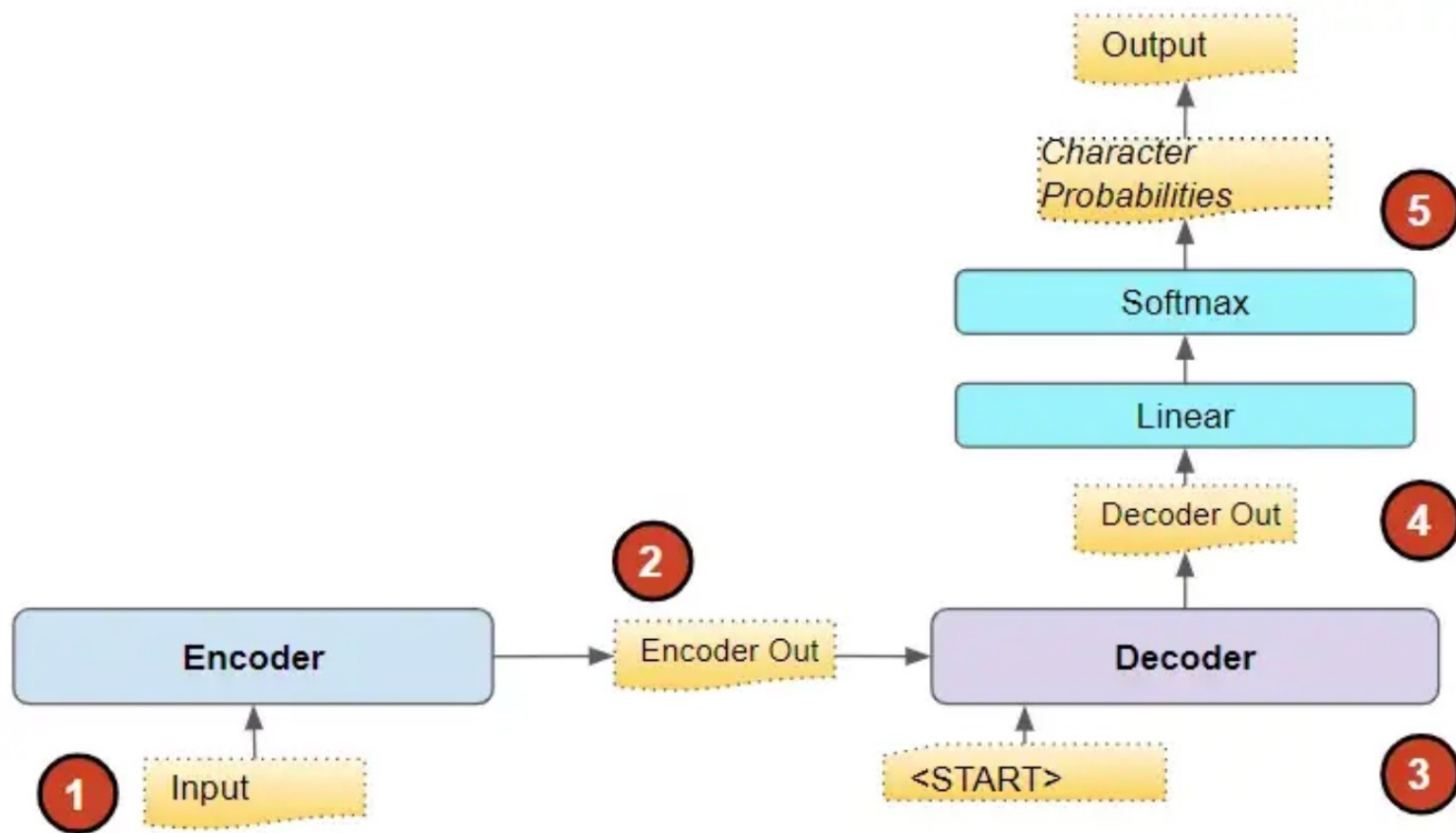
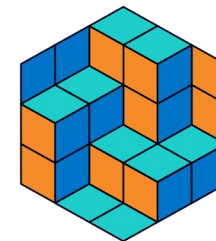


sees the whole text, but
something is corrupted

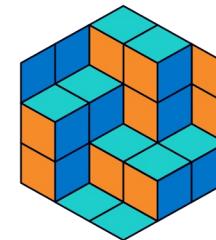
Глава 2

Greedy Search

Summary



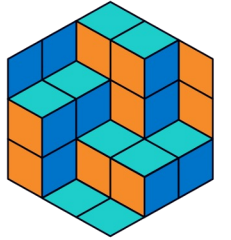
Summary



Vocab	A	0.12	0.09	...	0.82
	B	0.05	0.07

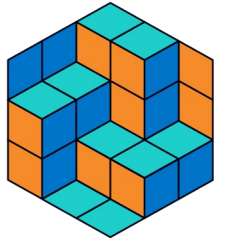
	Z	0.03
	END	0.01
Position:		1	2	...	n

Как выбрать выходной токен?



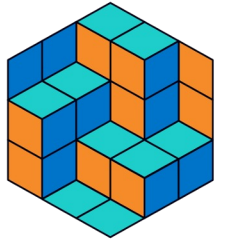
Greedy search – жадный алгоритм, а значит работает, но не всегда хорошо

Как выбрать выходной токен?

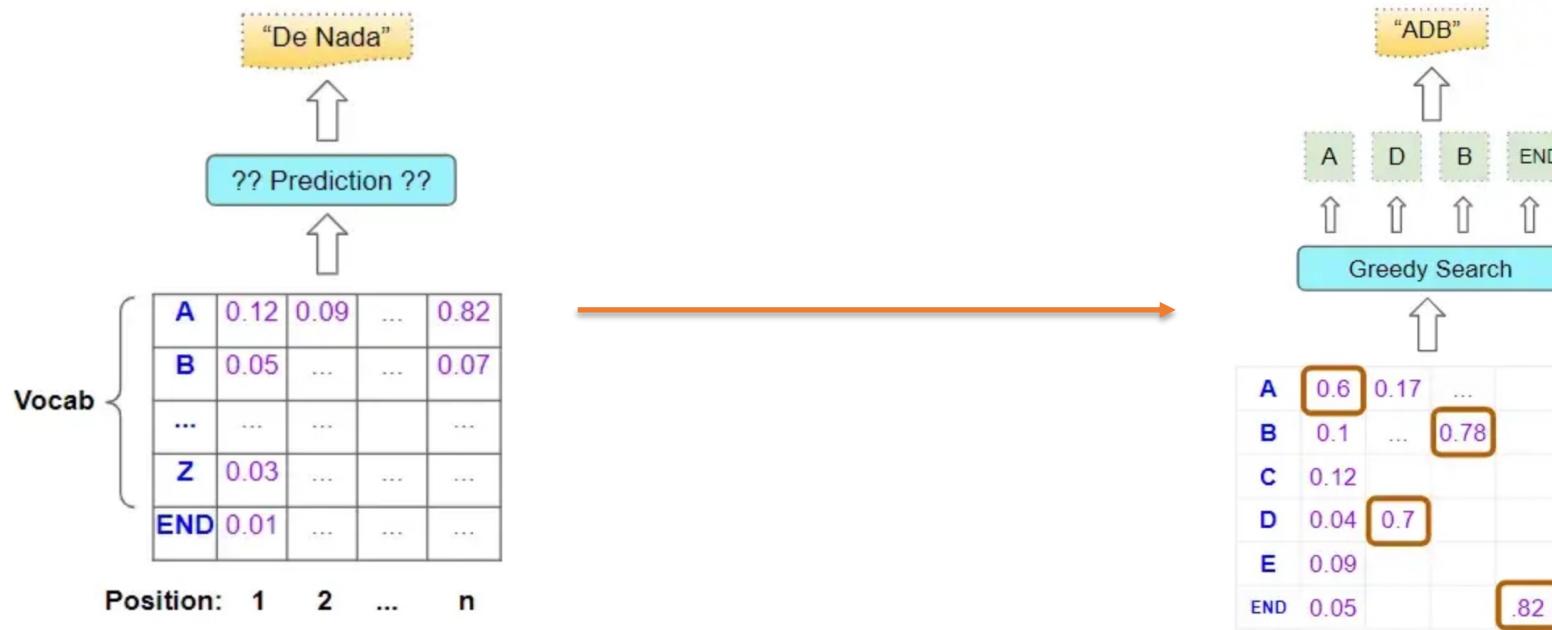


Greedy search – жадный алгоритм, а значит работает, но не всегда хорошо

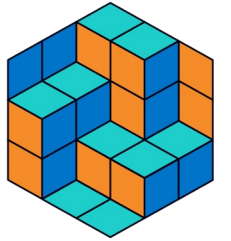
Как выбрать выходной токен?



Greedy search – жадный алгоритм, а значит работает, но не всегда хорошо



Как выбрать выходной токен?



Минусы:

- Не смотрит на контекст
- Часто зацикливается

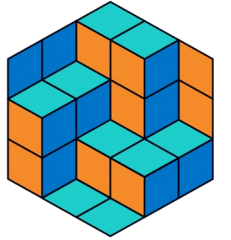
Плюсы:

- Быстро

Глава 3

Beam Search

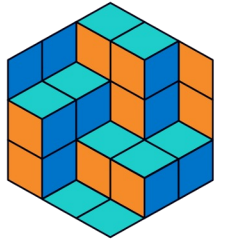
Как выбрать выходной токен? Pt. 2



Описание beam search:

1. На каждом шаге генерации выбираем top-N самых вероятных семплов

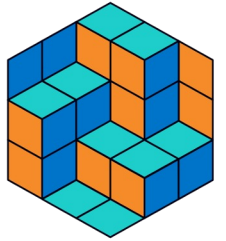
Как выбрать выходной токен? Pt. 2



Описание beam search:

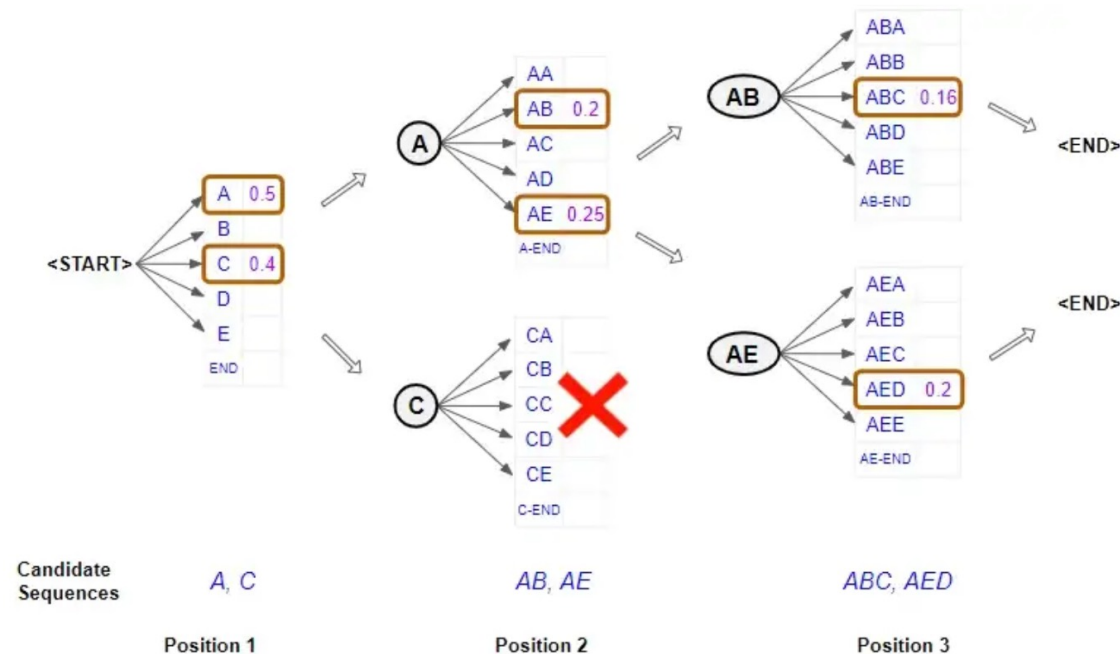
1. На каждом шаге генерации выбираем top-N самых вероятных семплов
2. Идем слева направо, и при каждом выборе самых вероятных, перезапускаем всю цепочку для пересчета наиболее возможных последовательностей

Как выбрать выходной токен? Pt. 2

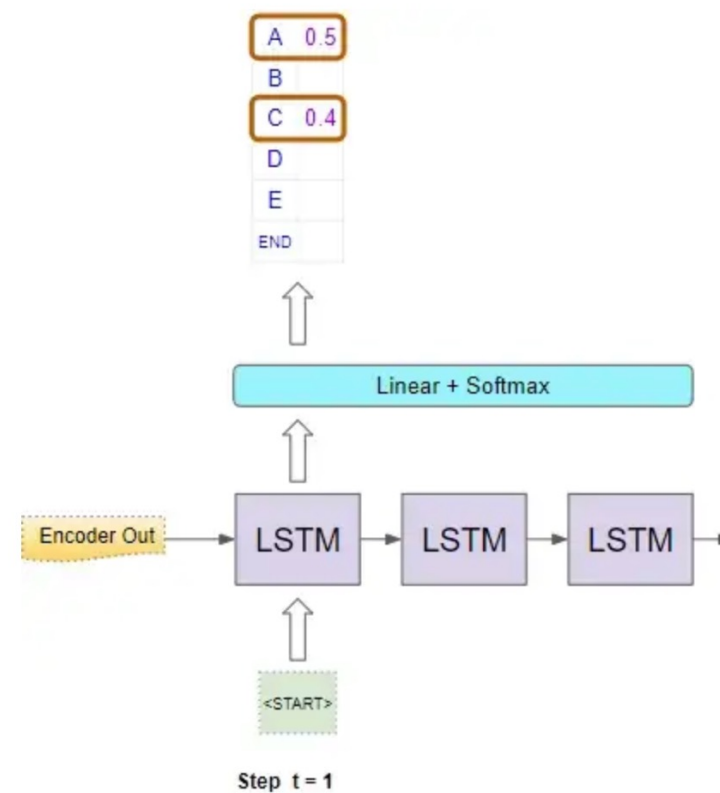
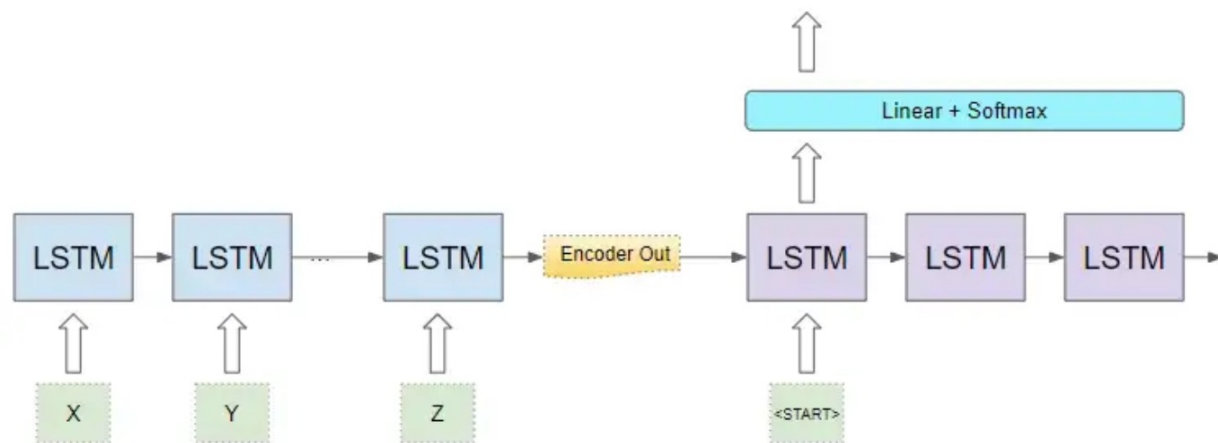
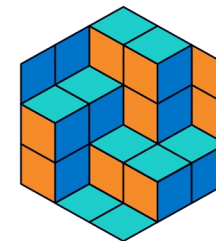


Описание beam search:

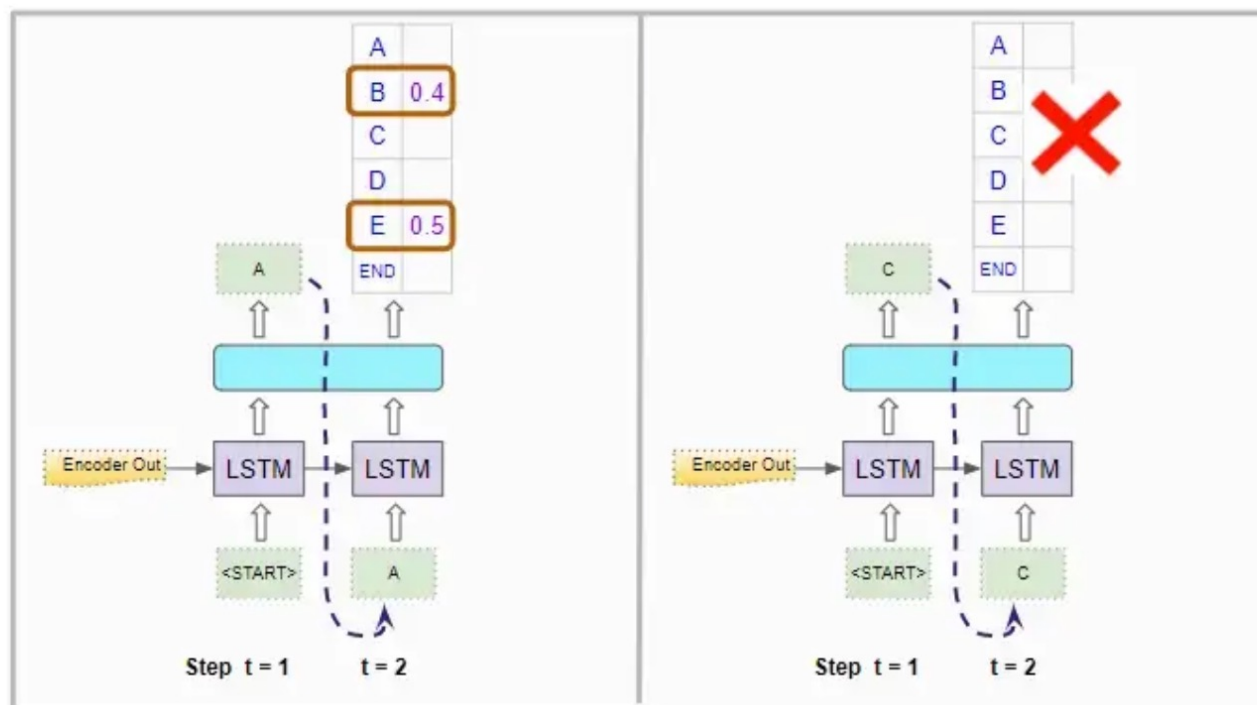
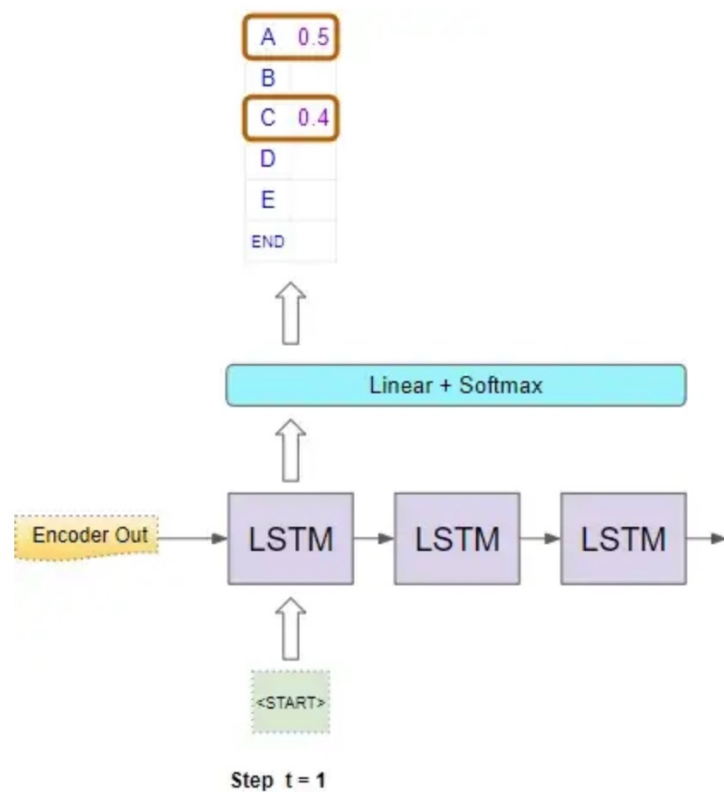
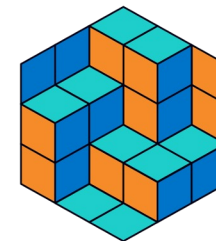
1. На каждом шаге генерации выбираем top-N самых вероятных семплов
2. Идем слева направо, и при каждом выборе самых вероятных, перезапускаем всю зепочку для пересчета наиболее возможных последовательностей



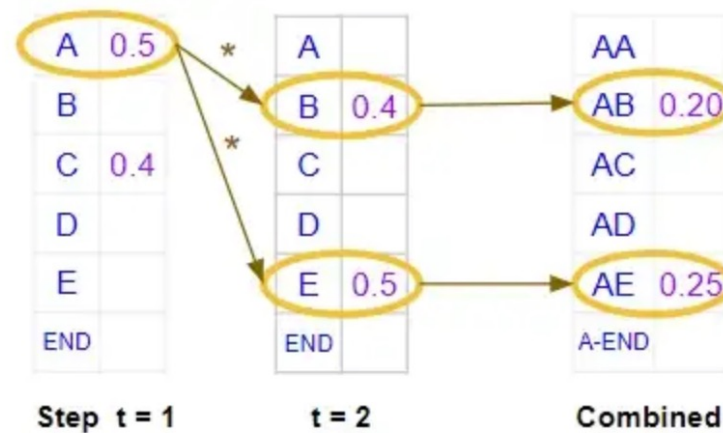
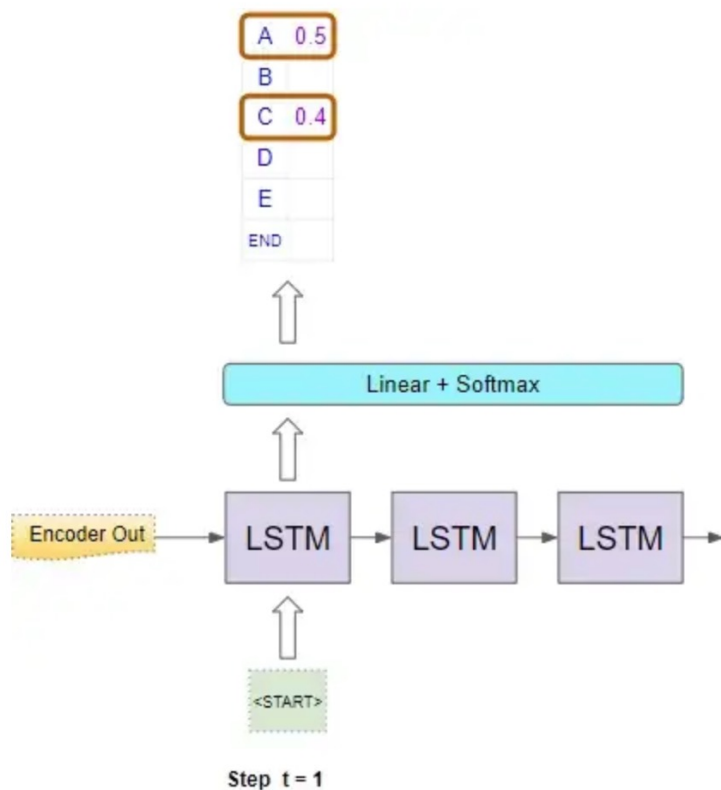
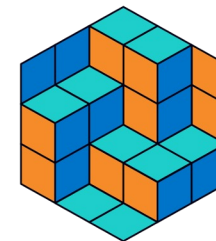
То есть?



То есть?



То есть?



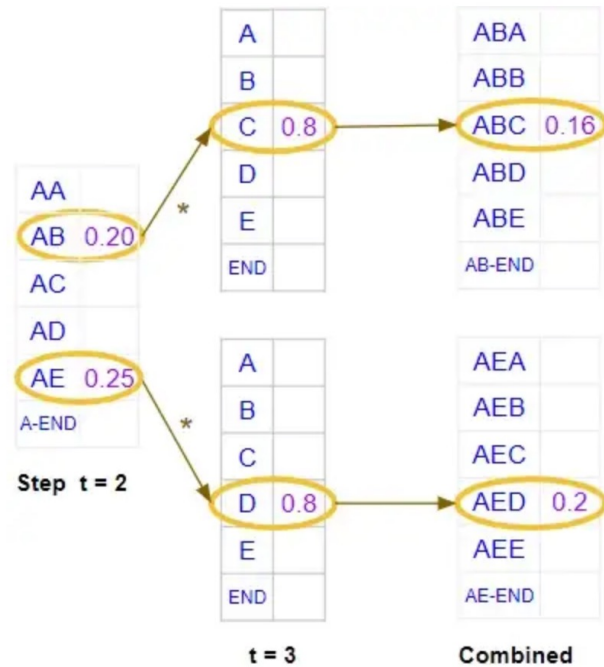
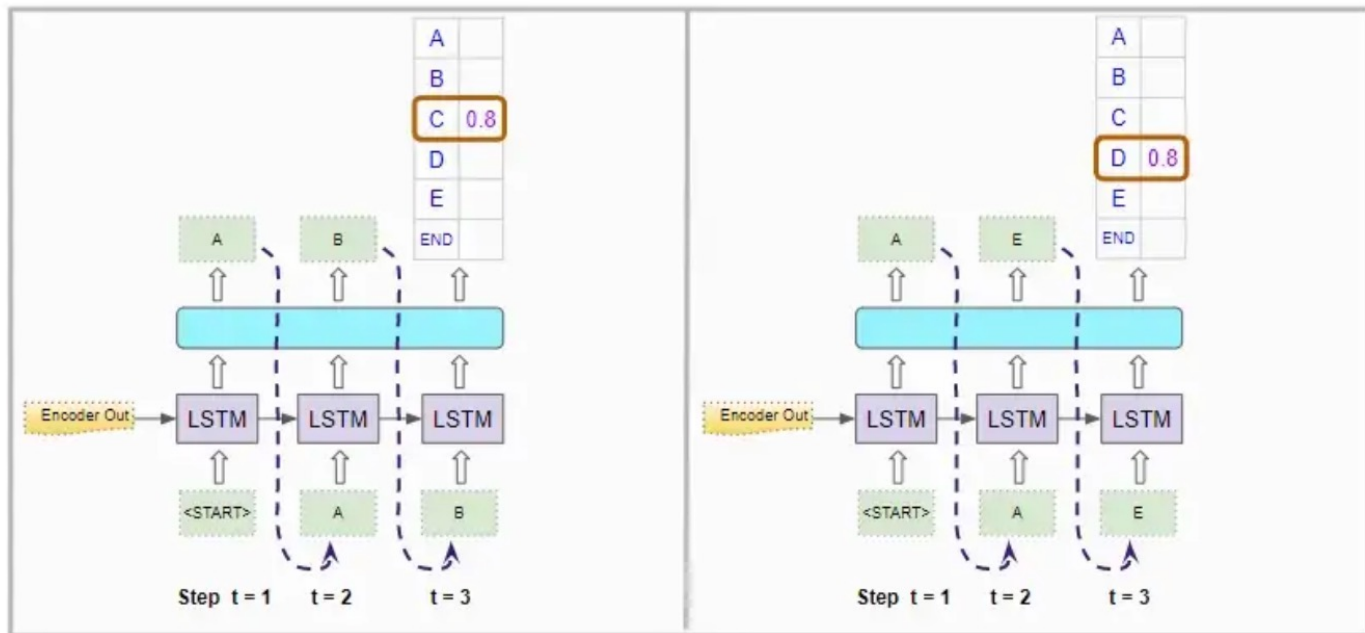
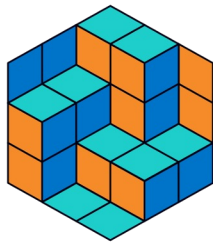
$$\text{Prob} (AB \mid \text{input}) = \text{Prob} (A \mid \text{input}) * \text{Prob} (B \mid A, \text{input})$$

$$\text{Prob} (AB) = \text{Prob} (A) * \text{Prob} (B \mid A)$$

$$= 0.5 * 0.4$$

$$= 0.20$$

То есть?



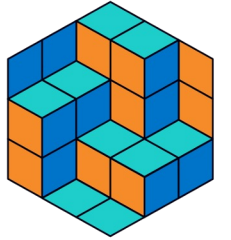
$$\text{Prob (ABC)} = \text{Prob (AB)} * \text{Prob (C | AB)}$$

$$= 0.2 * 0.8 = 0.16$$

$$\text{Prob (AED)} = \text{Prob (AE)} * \text{Prob (D | AE)}$$

$$= 0.25 * 0.8 = 0.2$$

Beam Search



Минусы:

- Долго

Плюсы:

- Генерирует не самые вероятные семплы, а самые вероятные последовательности

Вопросы?

