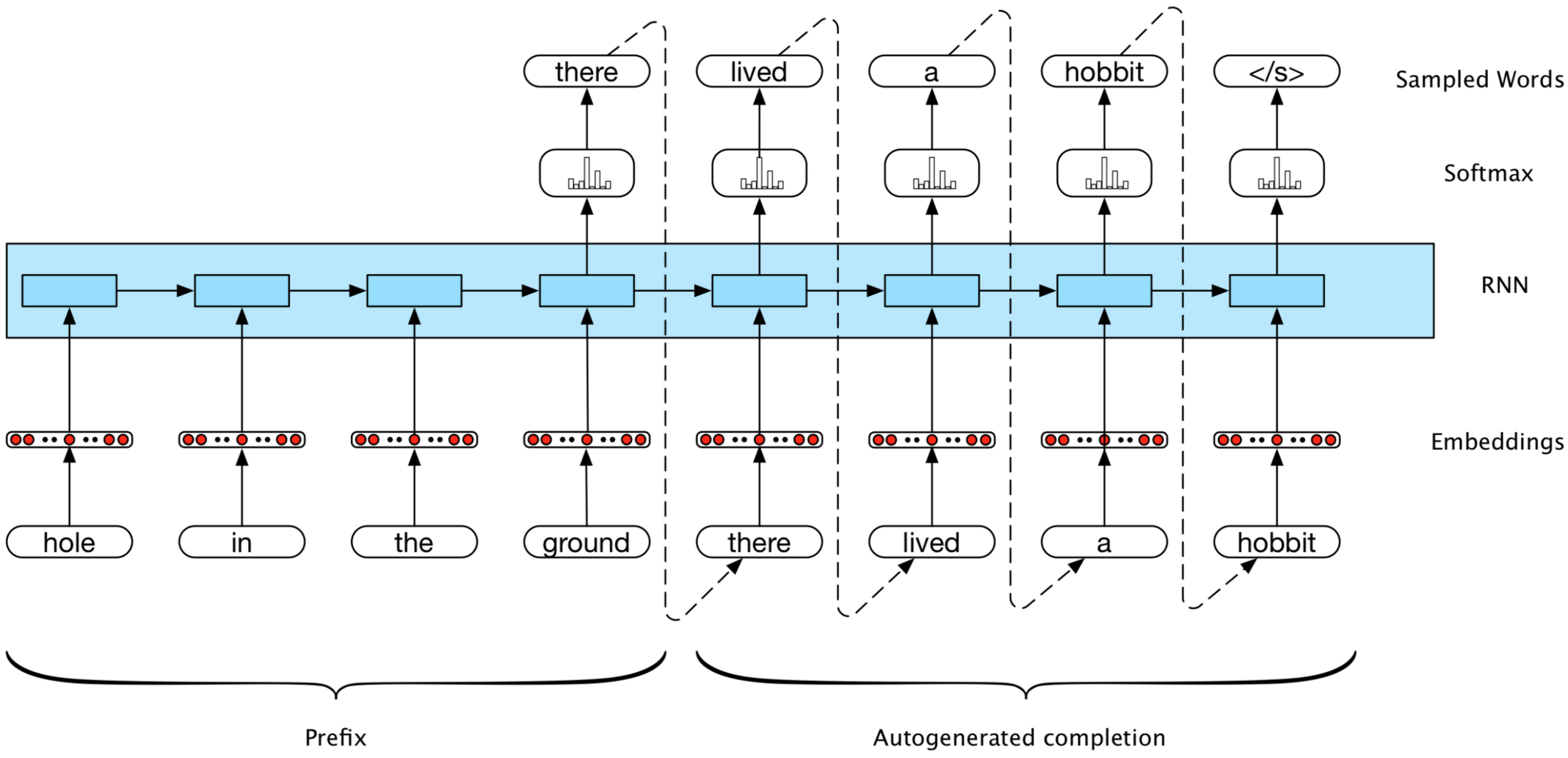
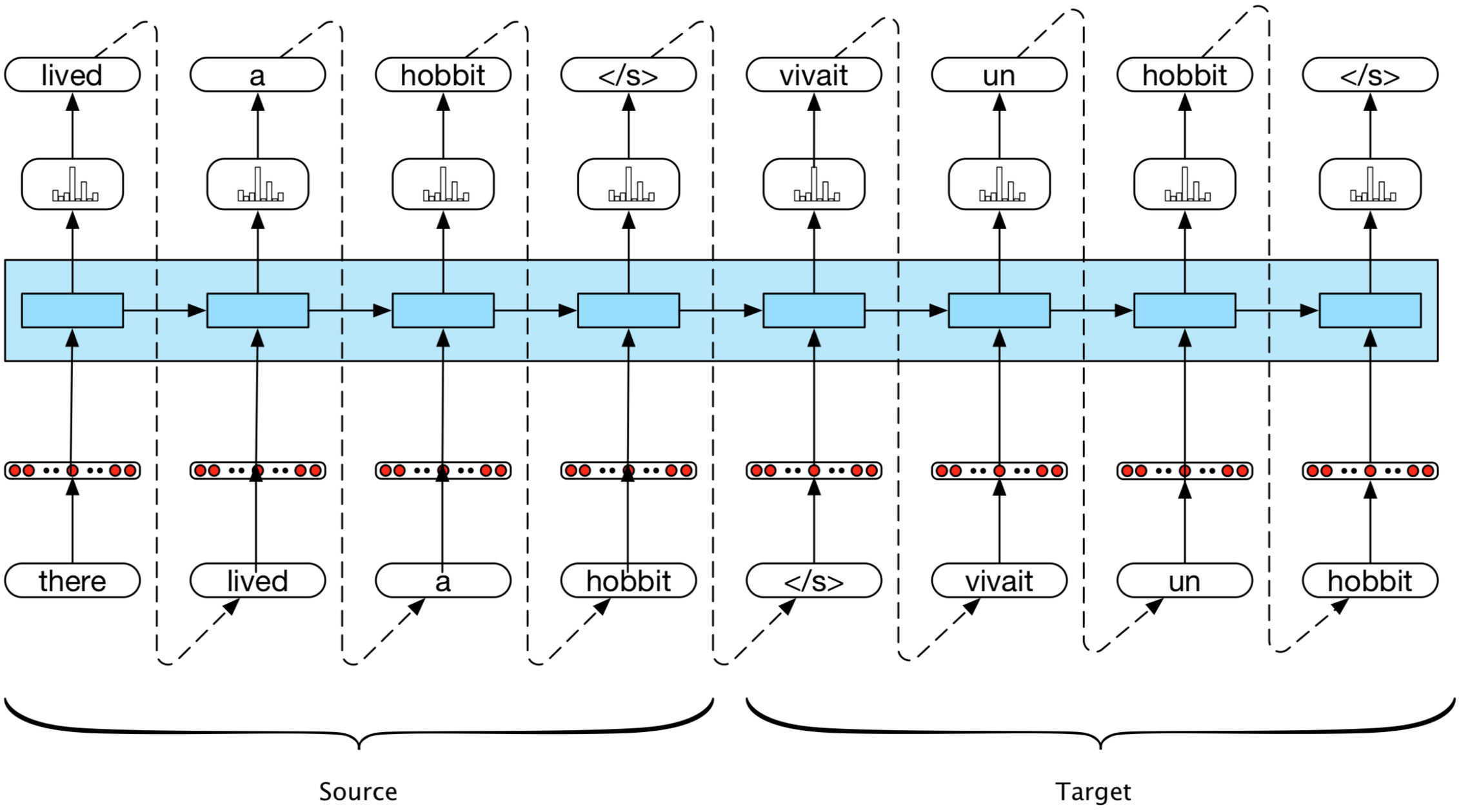
++RNN
Что делать дальше?

Языковые модели

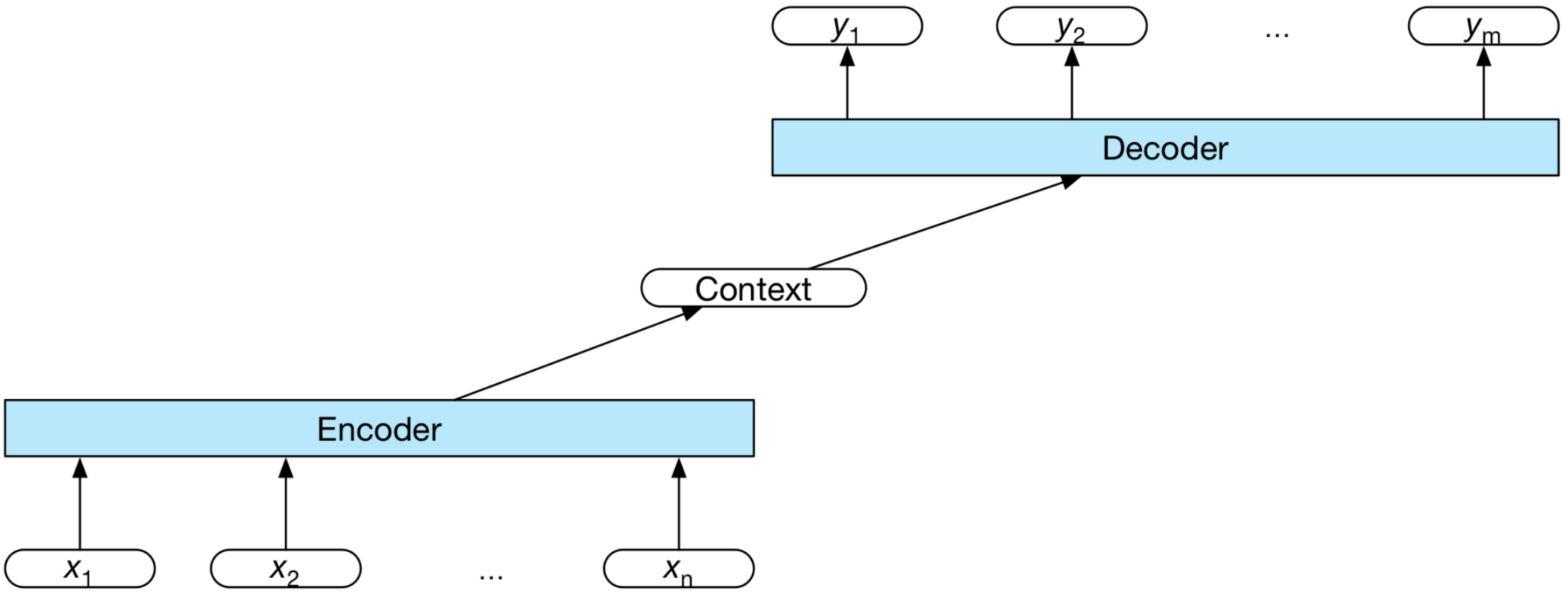


https://web.stanford.edu/~jurafsky/slp3/10.pdf

Машинный перевод



Encoder-Decoder архитектура



Проблемы:

- **Потеря информации при переходе от энкодера к декодеру**
- **Последние токены (слова) в энкодере всегда имеют больший вес**
- **Авторегрессивная манера генерации теряет связь с энкодером уже через несколько сгенерированных слов**

Механизм внимания

- **Предположение: а что если смотреть на все скрытые состояния сразу в момент генерации новых токенов?**

$$score(h^d_{i-1}, h^e_j) = h^d_{i-1} \cdot h^e_j$$ **- скалярное произведение**

$$score(h^d_{i-1}, h^e_j) = h^d_{t-1} W_s h^e_j$$ **- параметризация меры похожести**

- **Проблема: скоры имеют разную область значений. Что с этим делать?**

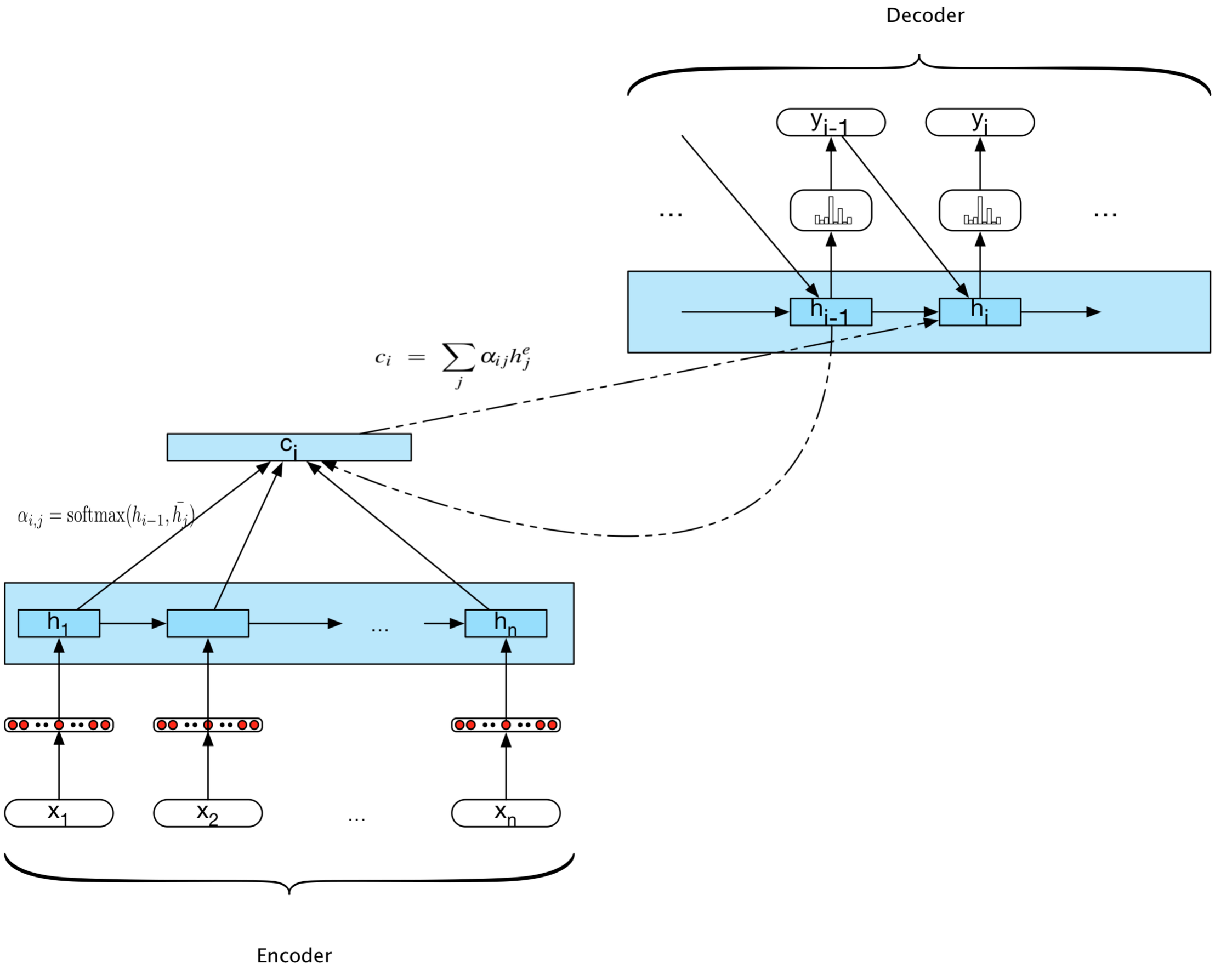
Нормализуем!

$$\alpha_{ij} = \text{softmax}(score(h^d_{i-1}, h^e_j) \ \ \forall j \in e)$$

$$= \frac{exp(score(h^d_{i-1}, h^e_j)}{\sum_k exp(score(h^d_{i-1}, h^e_k))}$$

Mex

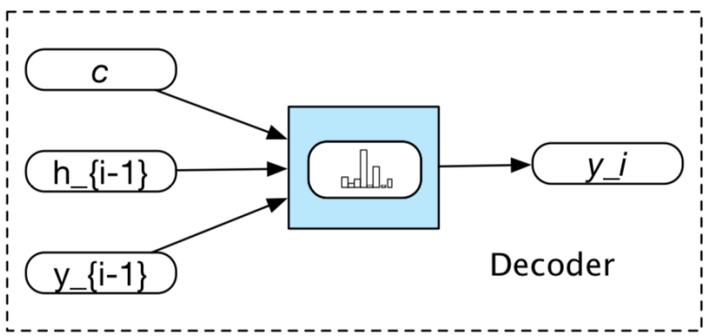
Для к

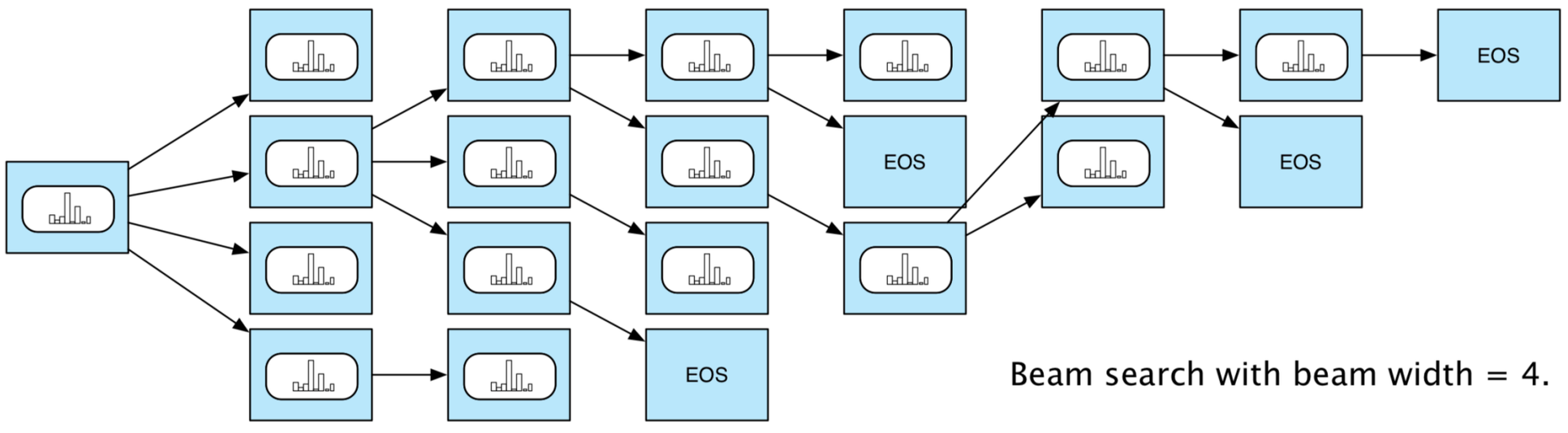

Decoder
y_{i-1}
y_i
...
...
h_{i-1}
h_i
$c_i = \sum_j \alpha_{ij} h_j^e$
c_i
$\alpha_{i,j} = \text{softmax}(h_{i-1}, \bar{h}_j)$
h_1
...
h_n
x_1
x_2
...
x_n
Encoder

Beam Search

- **Проблема: softmax может генерировать несколько хороших гипотез в декодере. Какую выбрать?**

- **Ответ: все сразу!**





Выводы:

- **Механизм внмимания**
- **Последние токены (слова) в энкодере всегда имеют больший вес**
- **Авторегрессивная манера генерации теряет связь с энкодером уже через несколько сгенерированных слов**

Оставшиеся проблемы:

- **Последовательная природа RNN**
- **(still) Длинные зависимости между словами и учитывание контекста**

Еще немного про зависимости

В машинном переводе выделяют три типа зависимостей:

- **Зависимость между оригиналом и переводом**
- **Зависимость между токенами оригинала**
- **Зависимость между токенами перевода**

Механизм внимания решил эту проблему

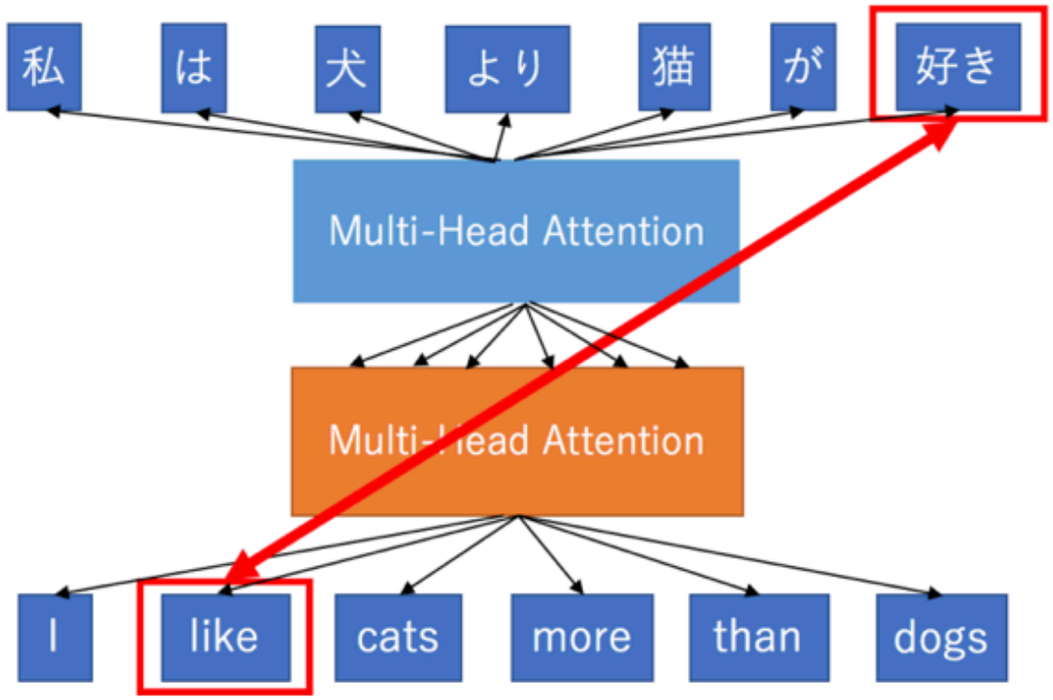
Осталось еще две

Attention is all you need

Новая статья предложившая решение данной проблемы:

- **Предположение 1: Давайте будем давать на вход энкодеру весь вход сразу и он сам решит, что важно а что нет**
- **Предположение 2: Механизм внимания поможет нам смоделировать эти зависимости**

Attention is all you need



The dependency that the Transformer has to learn. Now the path length is independent of the length of the source and target sentences.

http://mlexplained.com/2017/12/29/attention-is-all-you-need-explained/

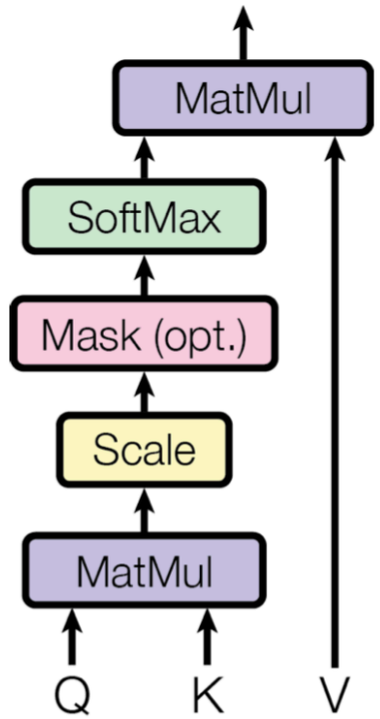
Scaled Dot-Product Attention

- **Очень простая функция (нейросеть), которая выдает скор attention для тройки Query (Q), Key (K), Value (V)**

- **Основана на Scaled Dot-Product Attention**

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

Scaled Dot-Product Attention

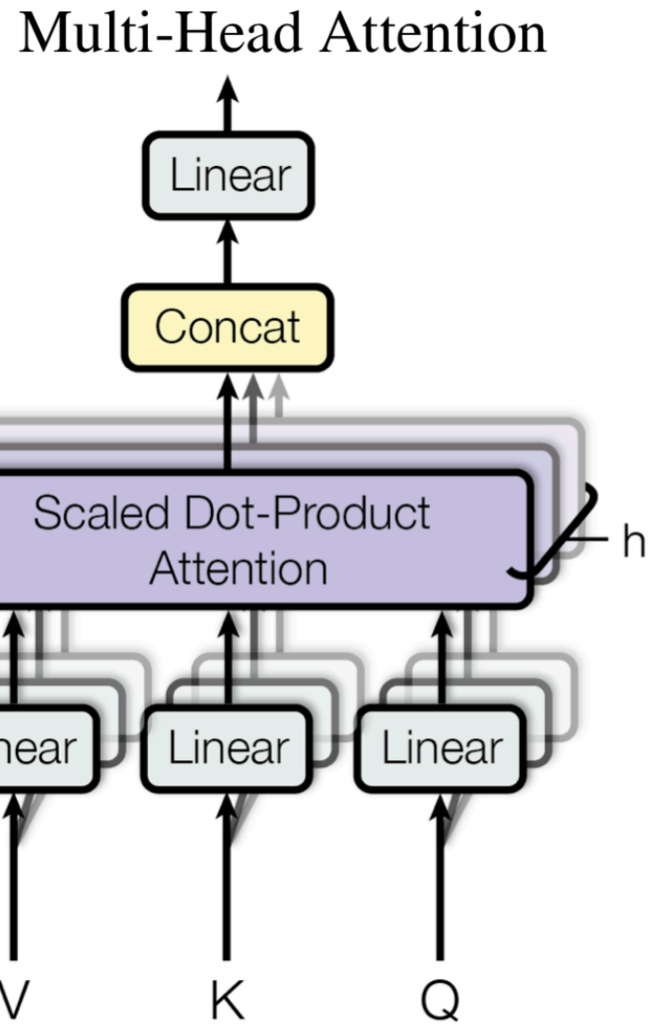


http://mlexplained.com/2017/12/29/attention-is-all-you-need-explained/

Multi-Head Attention

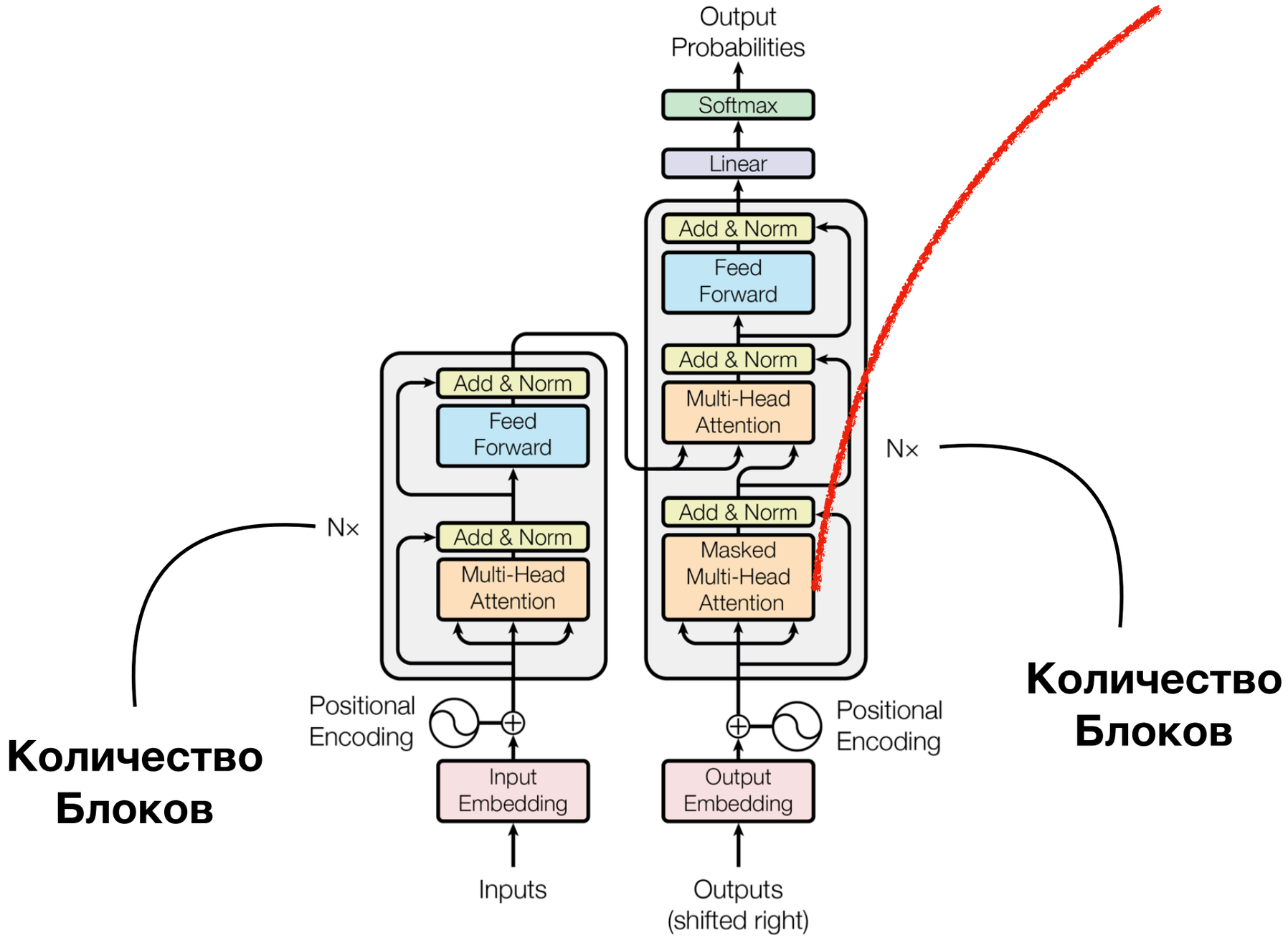
- **Сложная функция (нейросеть), которая выдает скор attention для тройки Query (Q), Key (K), Value (V)**

- **Основана на Scaled Dot-Product Attention**



Архитектура

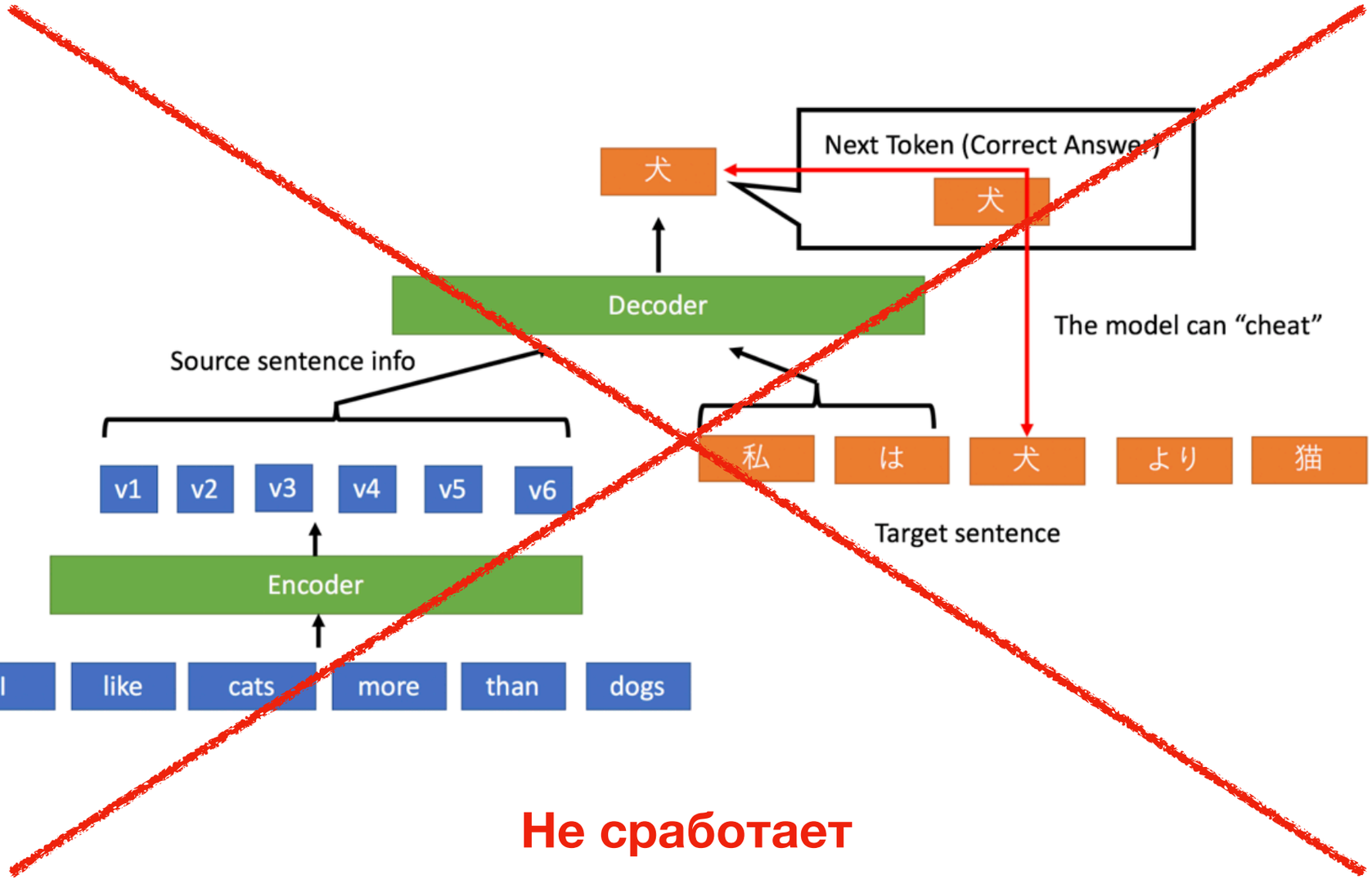
- **Encoder-Decoder сеть**



Masked Multi-Head Attention

- **Модификация предыдущего слоя для работы с декодером**

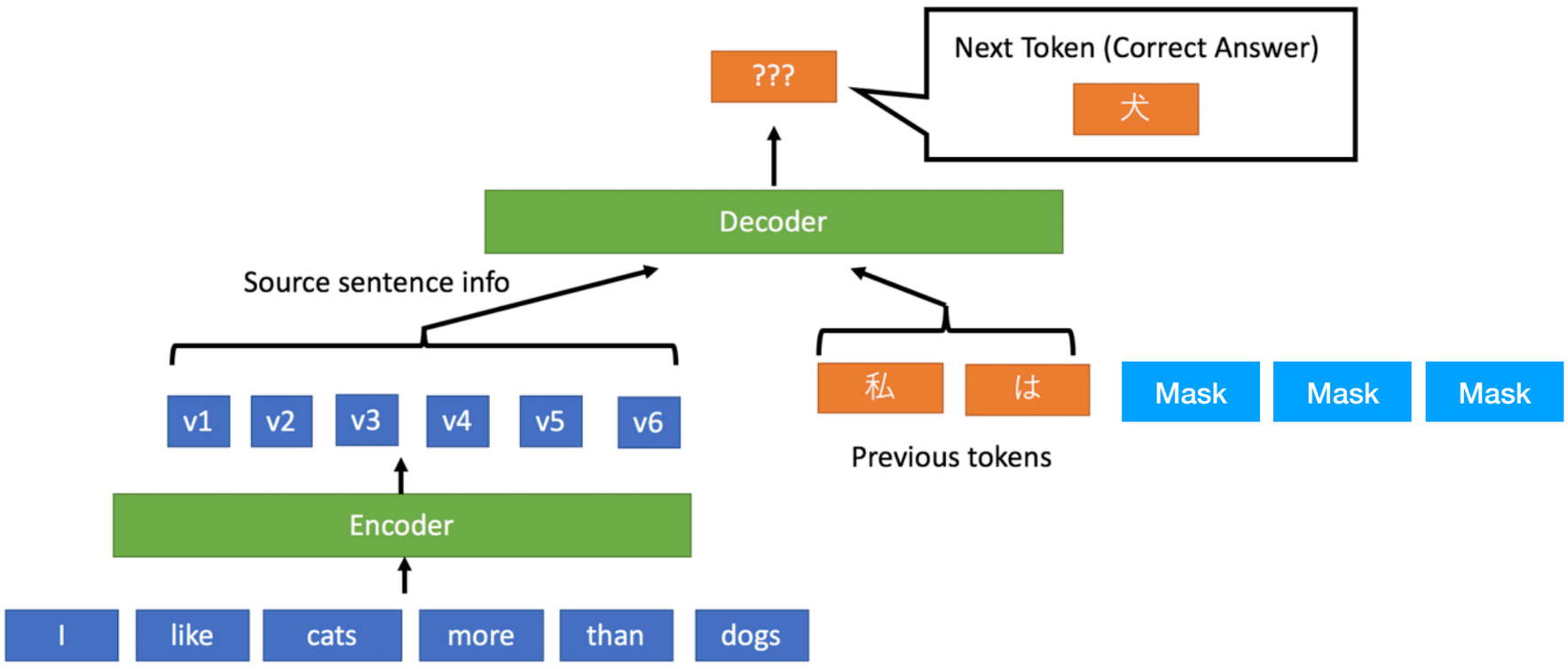
- **Изначально, мы говорили, что хотим работать со всей последовательностью токенов. Проблема: сеть не выучит закономерностей и сломается на inference этапе**



Не сработает

Masked Multi-Head Attention

Решение: Будем прятать от декодера будущие токены



Positional Embeddings

- **Каждый Multi-Head Attention это полносвязная сеть.**
- **Проблема: не знаем ничего о позициях токенов**
- **Решение: Авторы предложили добавить информацию о позиции прямо в эмбеддинги**

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

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

- **Очередной способ получить эмбеддинги**
- **Зачем?**
- **Ответ:**
  1. **W2V не может получать закономерности в нескольких эмбеддингах сразу.**
  2. **Мы не можем получить сразу эмбеддинг для предложения**

http://mlexplained.com/2019/01/07/paper-dissected-bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding-explained/

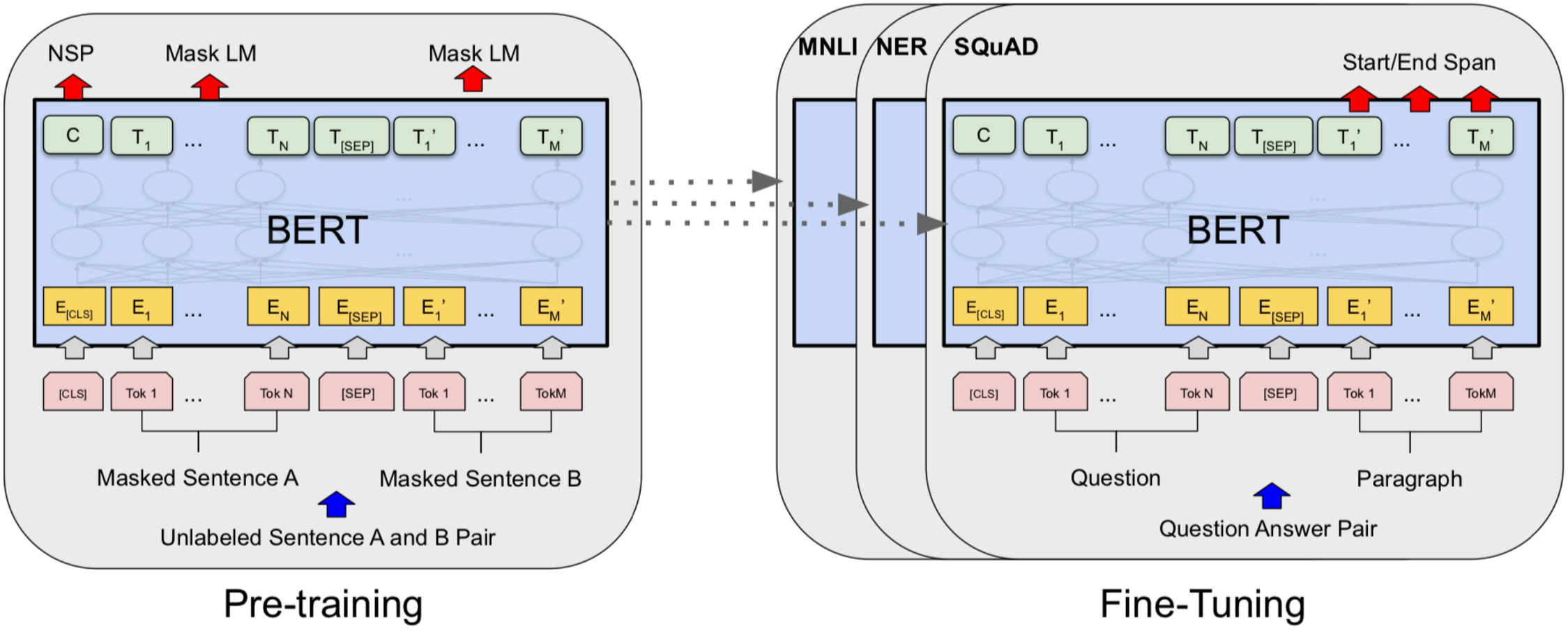
BERT

- **Тренируется иначе и на других задач**
- **Использует трансформер для энкодинга предложений**
- **Работает лучше**

Как тренируется BERT?

- **Использует subword units**
- **Решаем задачу языкового моделирования**
- **Рандомно закрываем слова в предложении и предсказываем их**

Еще одна полезная картинка



Pre-training

Fine-Tuning

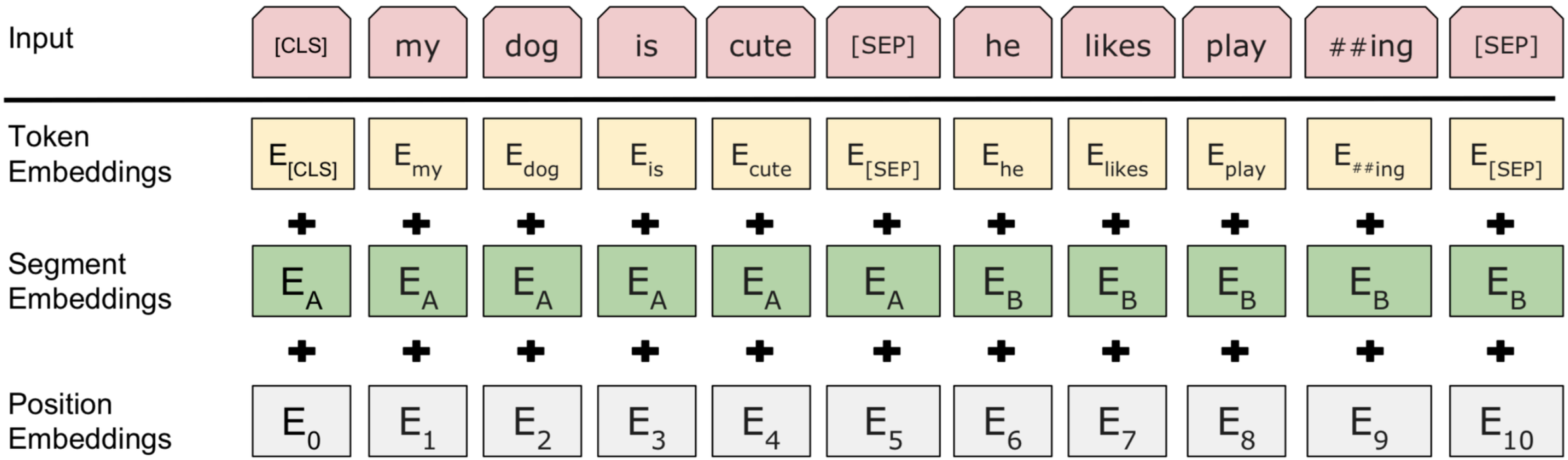
Как тренируется BERT?

- **Использует subword units**
- **Предсказывает являются ли предложения продолжением друг друга**

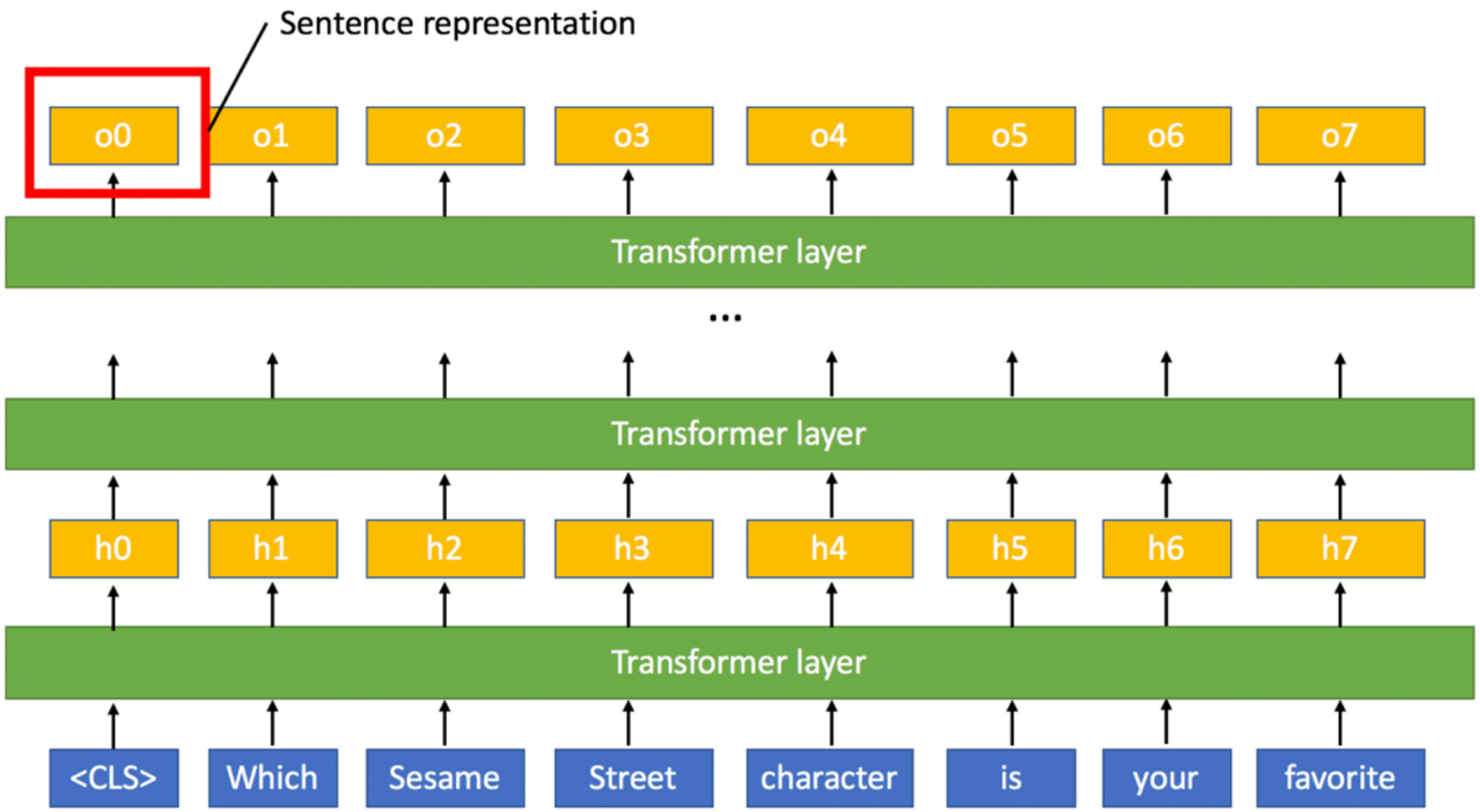
$$\text{Input} = \texttt{[CLS] the man went to [MASK] store [SEP]}$$
$$\texttt{he bought a gallon [MASK] milk [SEP]}$$
$$\text{Label} = \texttt{IsNext}$$

$$\text{Input} = \texttt{[CLS] the man [MASK] to the store [SEP]}$$
$$\texttt{penguin [MASK] are flight ##less birds [SEP]}$$
$$\text{Label} = \texttt{NotNext}$$

Архитектура



Архитектура



Материалы

- https://web.stanford.edu/~jurafsky/slp3/10.pdf
- http://mlexplained.com/2017/12/29/attention-is-all-you-need-explained/
- https://arxiv.org/pdf/1706.03762.pdf
- http://mlexplained.com/2019/01/07/paper-dissected-bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding-explained/
- https://arxiv.org/pdf/1810.04805.pdf