MIPT
MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

АТП кафедра Алгоритмов и Технологий Программирования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Автоматическая обработка текстов Seq2Seq и Attantion

Лекция 6

Емельянов А. А.
login-const@mail.ru

2018

https://github.com/king-menin/nlp-hse-autumn2018

Задача машинного перевода (Seq2Seq)

- Исходное предложение: $x = (x_1, x_2, \dots, x_{T_x})$
- Целевое предложение: $y = (y_1, y_2, \dots, y_{T_y})$
- Любой тип системы машинного перевода может быть представлен как функция:

$$\hat{y} = mt(x)$$

- Перевод эквивалентен поиску целевого предложения, которое максимизирует условную вероятность y при известном x, т. е.

$$argmax_y p(y|x)$$

Задача машинного перевода (Seq2Seq)

- Системы машинного перевода создают вероятностную модель для вероятностей y при условии x,

$$p(y|x,\theta)$$

- и ищут целевое предложение максимизируя вероятность:

$$\hat{y} = argmax_y p(y|x,\theta)$$

- θ – это параметры модели, задающие за распределение вероятностей.

Задача машинного перевода (Seq2Seq)

- **Моделирование**:

Какую модель $p(y|x,\theta)$ мы ищем?

- **Обучение:**

Нам необходим метод для обучения подходящих параметров θ по тренировочным данным.

- **Поиск:**

Мы должны решить проблему поиска наиболее вероятного предложения (решение: «*argmax*»).

«Условные» языковые модели

- Seq2Seq – непосредственные модели вероятности $P(y|x)$:

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$

«Условные» языковые модели

- Seq2Seq – непосредственные модели вероятности $P(y|x)$:

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$

Вероятность следующих целевых слов до сих пор и исходное предложение x

Условные языковые модели

- Seq2Seq – непосредственные модели вероятности $P(y|x)$:

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$

Вероятность следующих целевых слов до сих пор и исходное предложение x

- Почему условные языковые модели?

Условные языковые модели

- Seq2Seq – непосредственные модели вероятности $P(y|x)$:

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots, P(y_T|y_1, \ldots, y_{T-1}, x)$$

Вероятность следующих целевых слов до сих пор и исходное предложение x

- Почему условные языковые модели?
  - Языковая модель предсказывает следующее слово?
  - Условная – предсказания обусловлены исходным предложением.

Encoder-Decoder архитектура

Recap: RNN LM

- Рекуррентные языковые модели (RNN LM)

- Итоговое распределение:

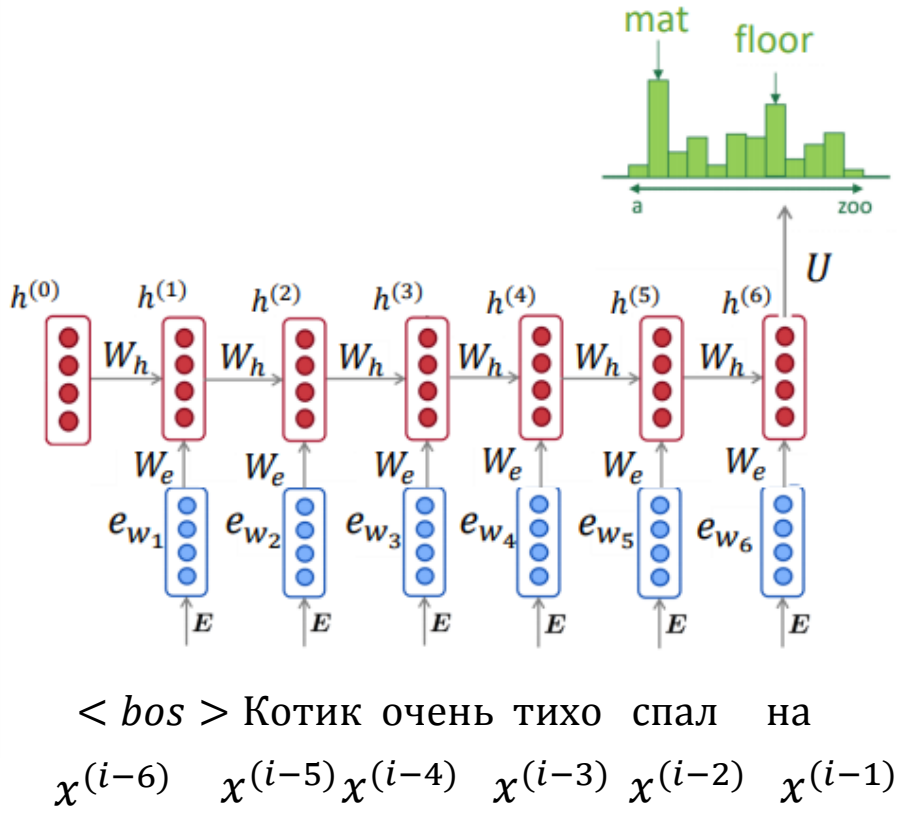
$$\hat{y}_t = softmax\left(Uh^{(t)} + b_2\right) \in R^{|V|}$$

- Скрытые состояния

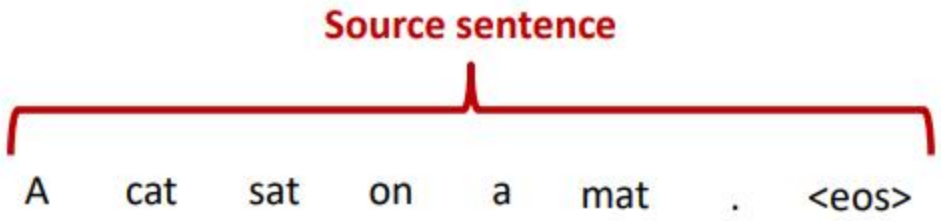
$$h^{(t)} = \sigma\left(W_h h^{(t-1)} + W_x e^{(t-1)} + b_1\right)$$

- Эмбеддинги слов

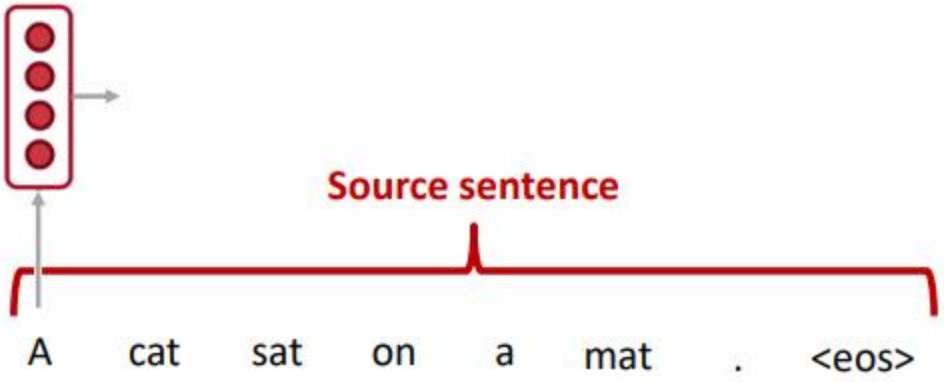
- Слова или токены



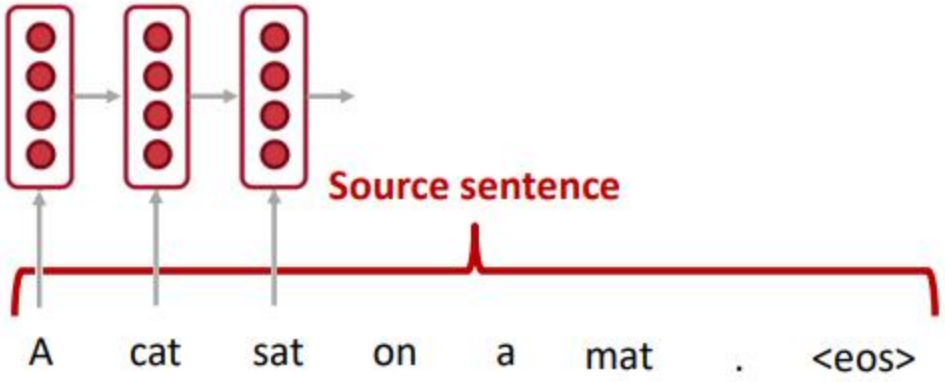
RNN Encoder-Decoder (Vanilla)



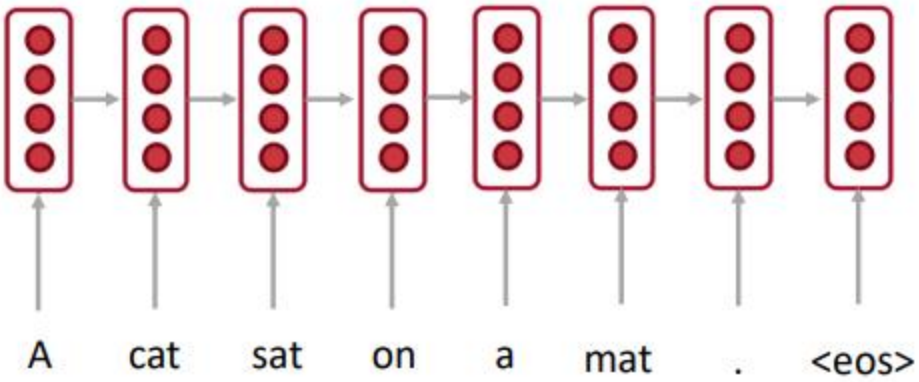
RNN Encoder-Decoder (Vanilla)



RNN Encoder-Decoder (Vanilla)

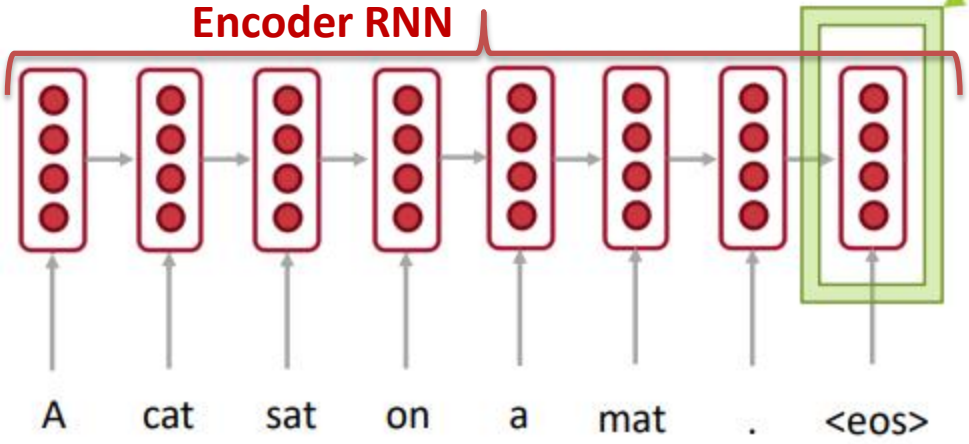


RNN Encoder-Decoder (Vanilla)



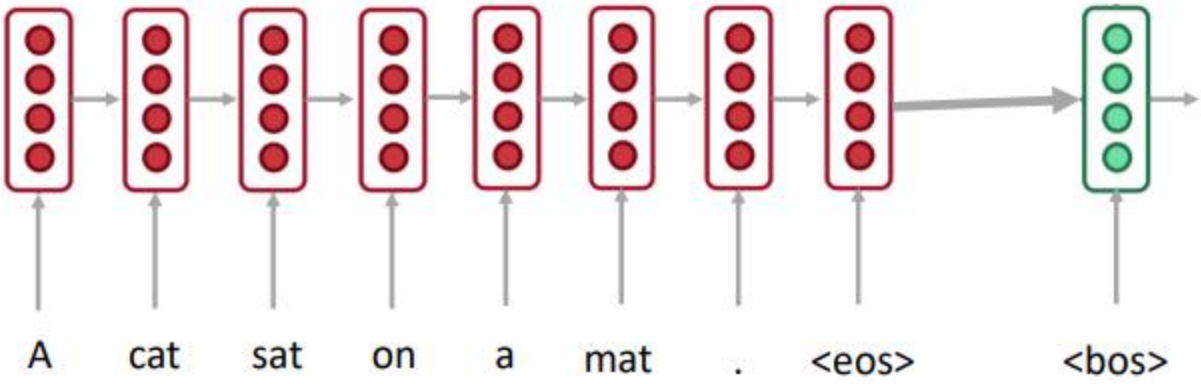
RNN Encoder-Decoder (Vanilla)

Encoder RNN

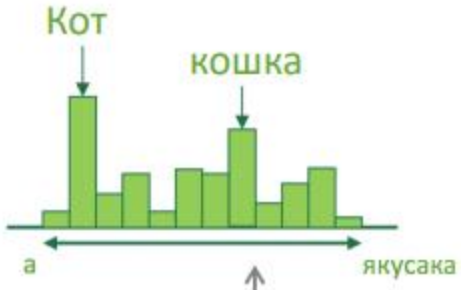


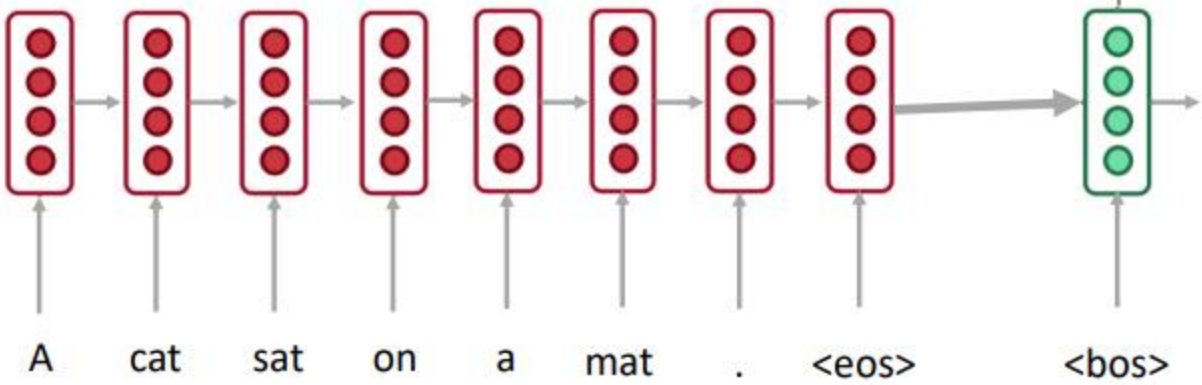
Закодированное исходное предложение. Используем в качестве исходного состояния для decoder RNN

RNN Encoder-Decoder (Vanilla)



RNN Encoder-Decoder (Vanilla)



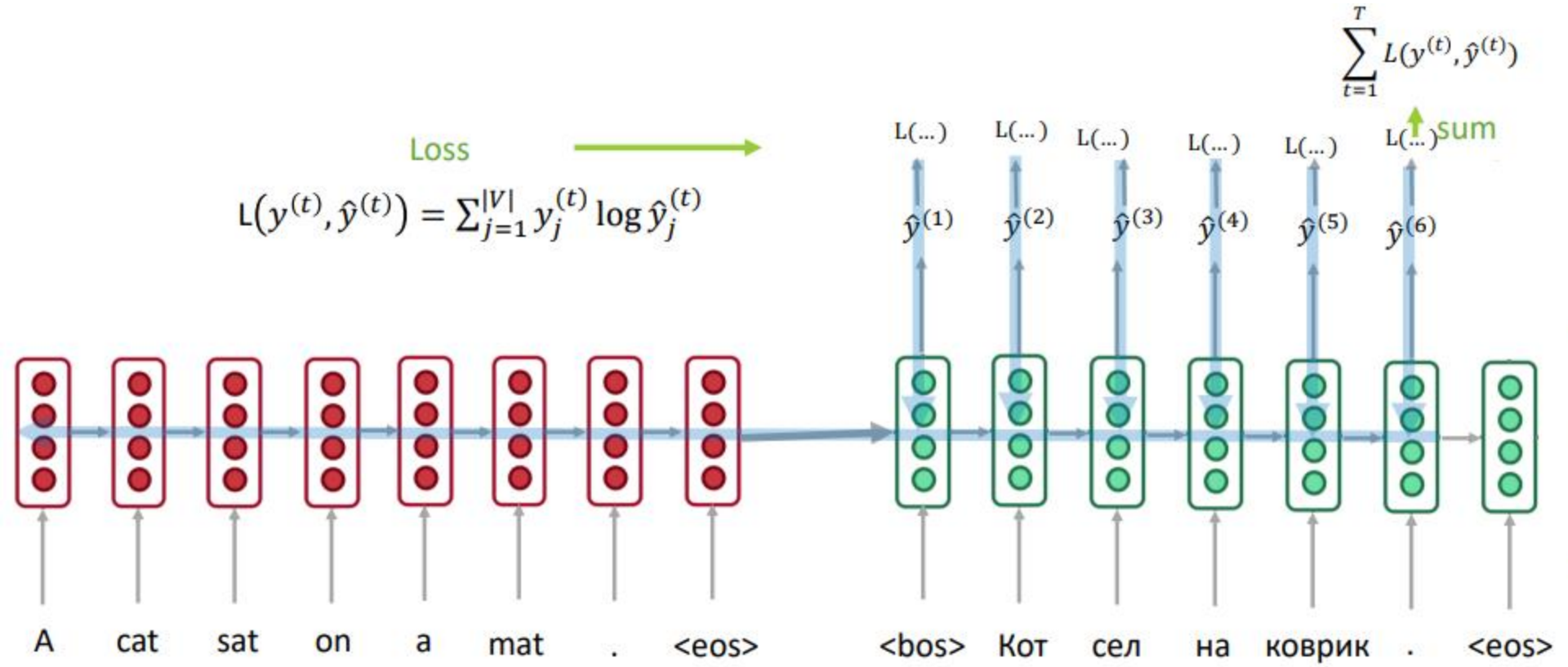


RNN Encoder-Decoder (Vanilla)

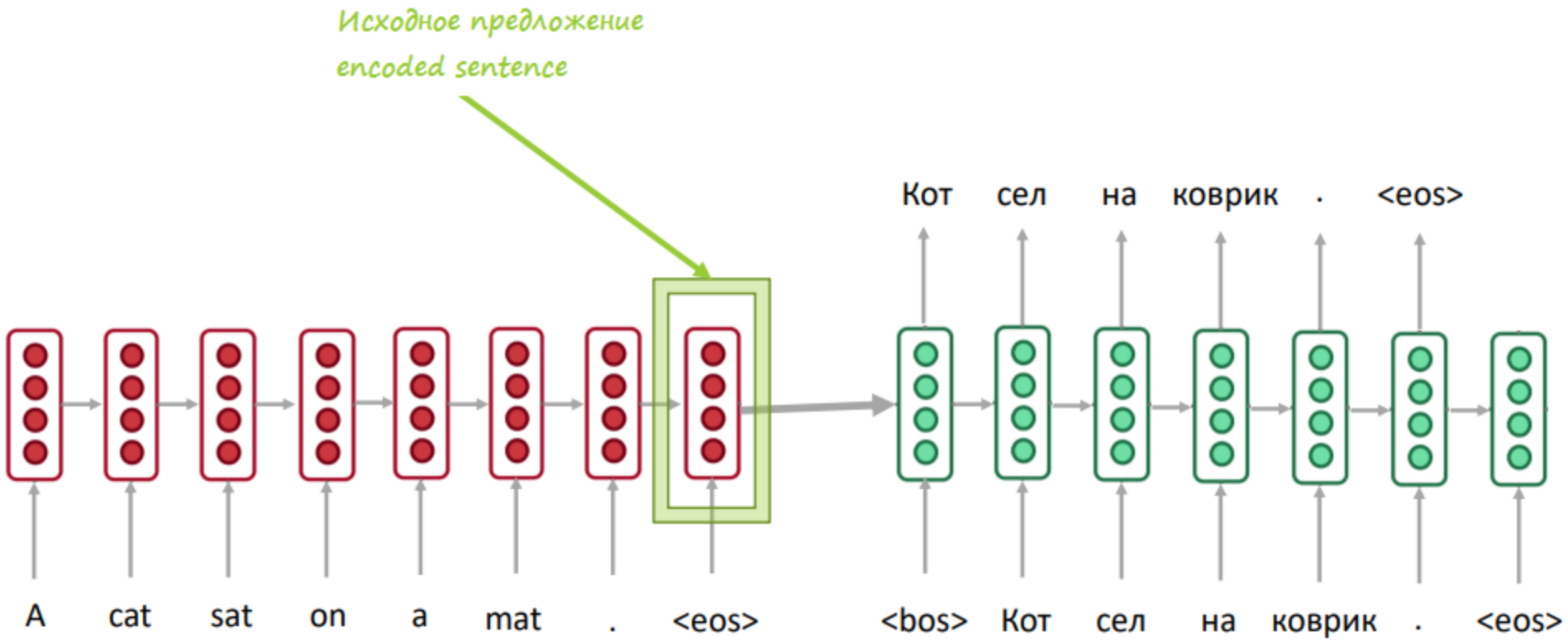
RNN Encoder-Decoder (Vanilla)

RNN Encoder-Decoder (Vanilla)

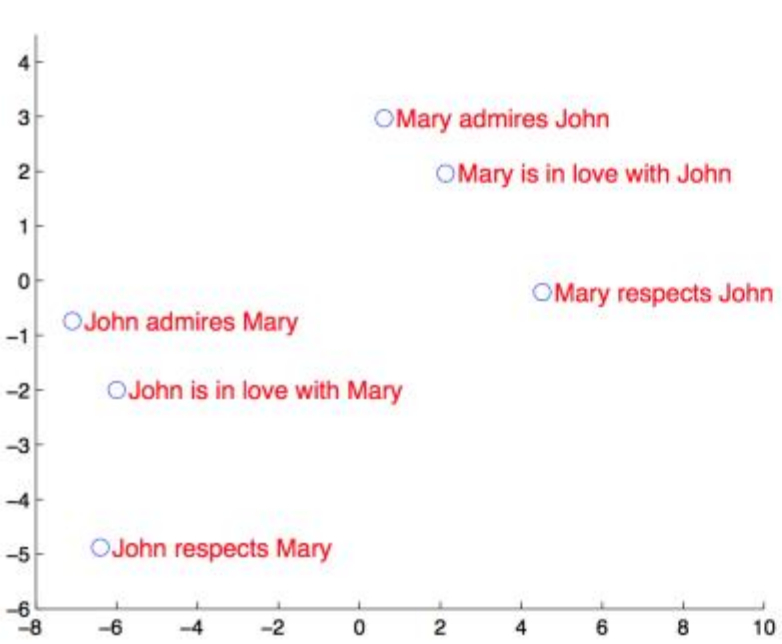
RNN Encoder-Decoder (Vanilla)

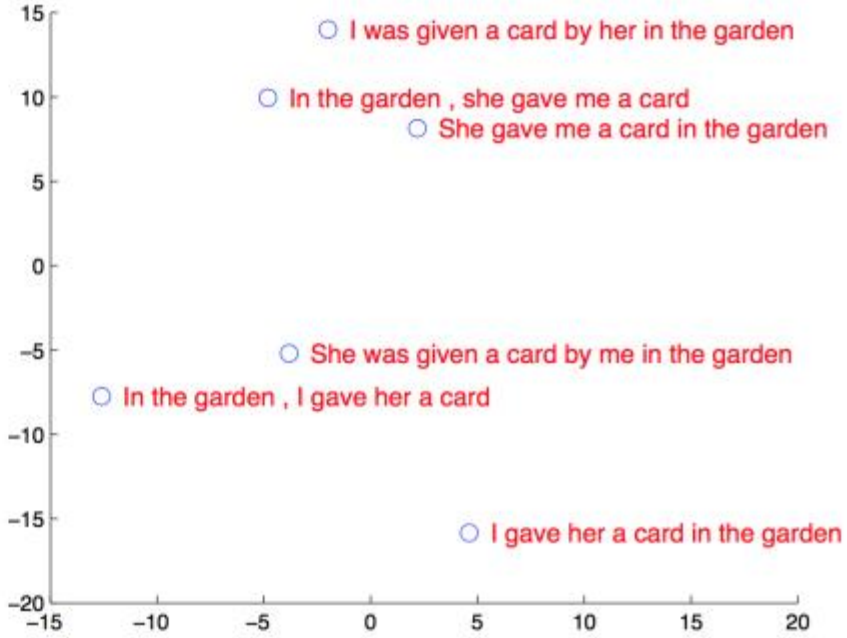


Посмотрим на эмбеддинги предложения после encoder



Посмотрим на эмбеддинги предложения после encoder



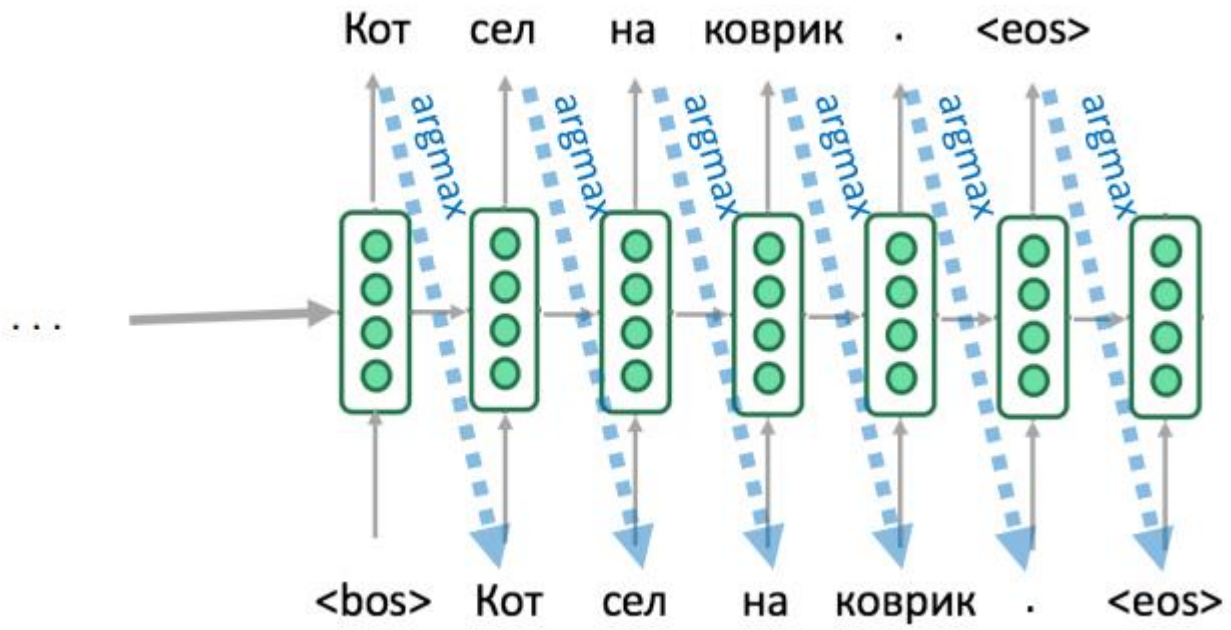


1. Sutskever et al, NIPS 2014, https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

Decoding: Как генерировать последовательность?

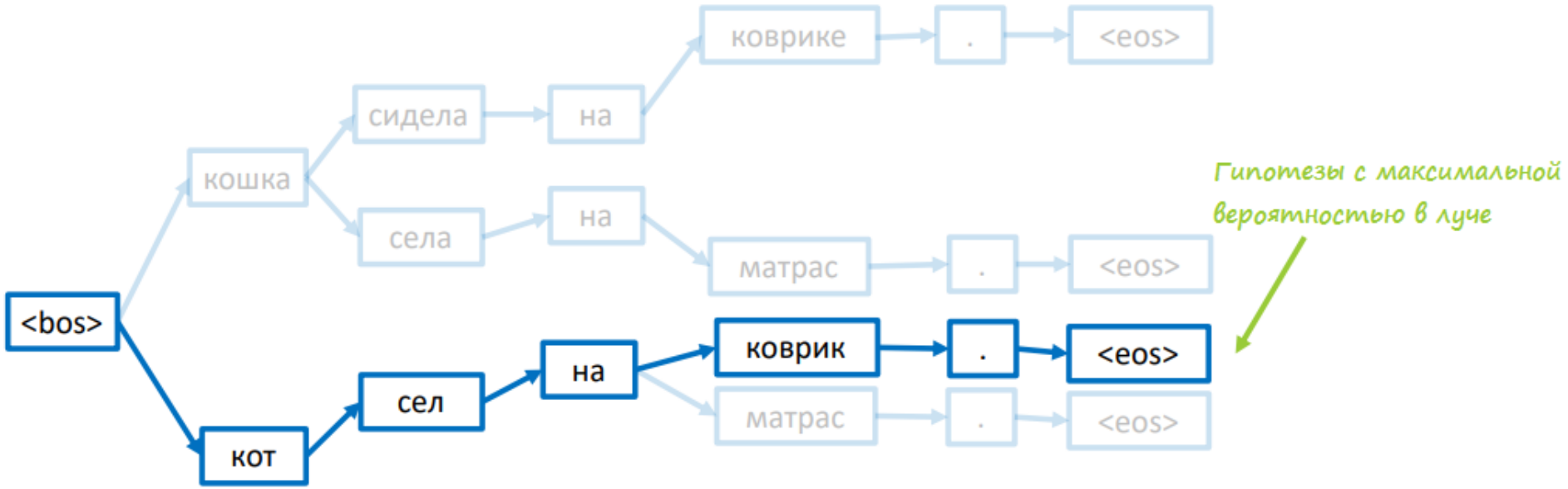
Decoding: Как генерировать последовательность?

- **Greedy decoding**

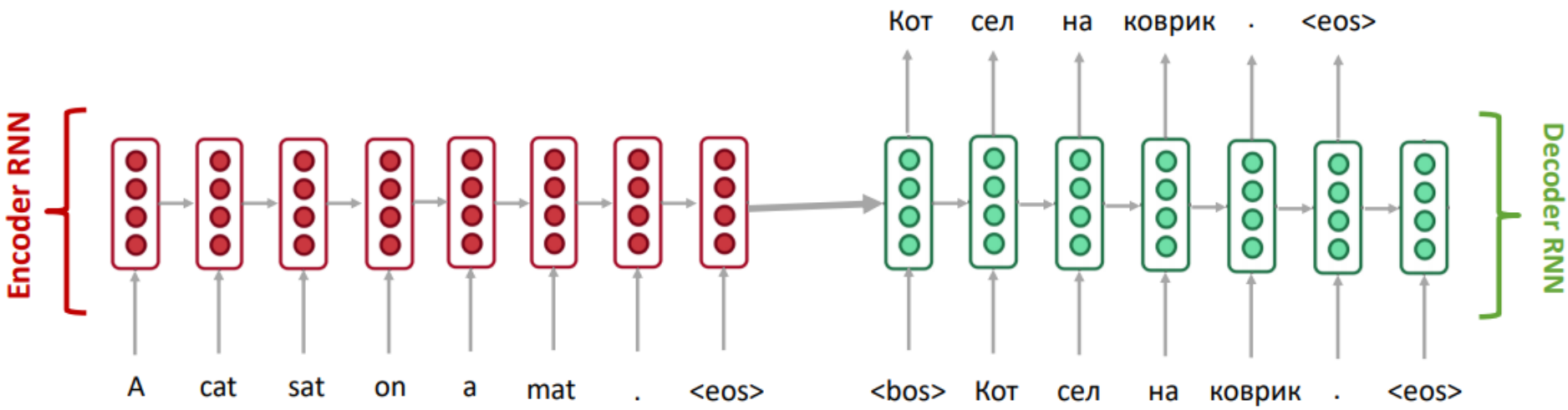


Decoding: Как генерировать последовательность?

- **Beam search**

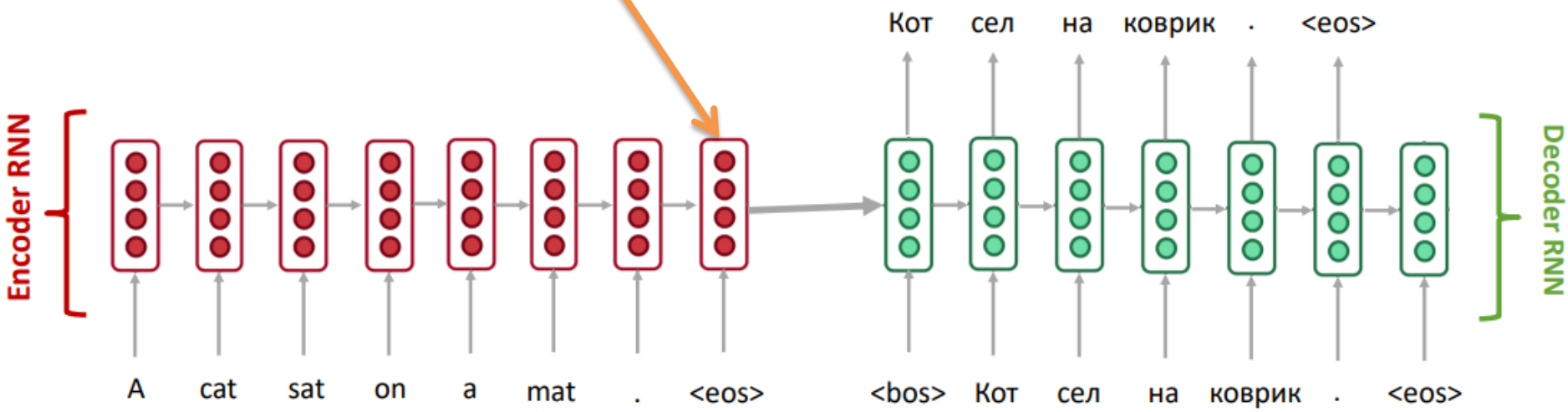


Какие проблемы с обычными RNN Encoder-Decoder моделями?

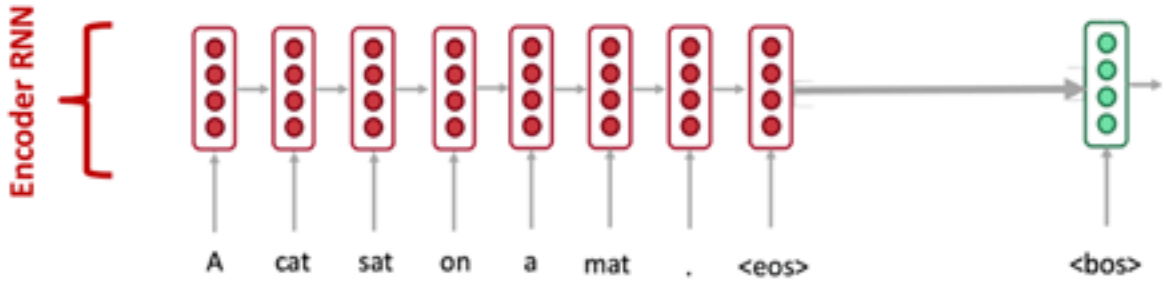


Какие проблемы с обычными RNN Encoder-Decoder моделями?

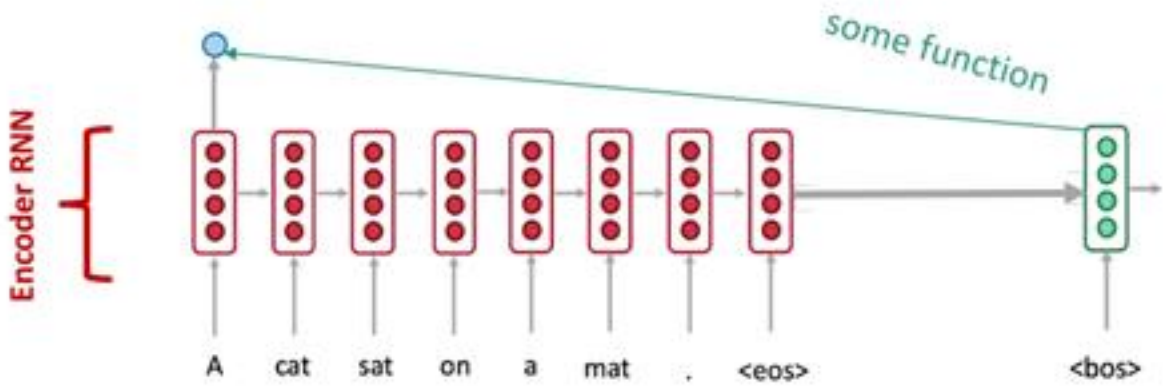
Information bottelneck:
модель должна помещать всю информацию о предложении в один вектор



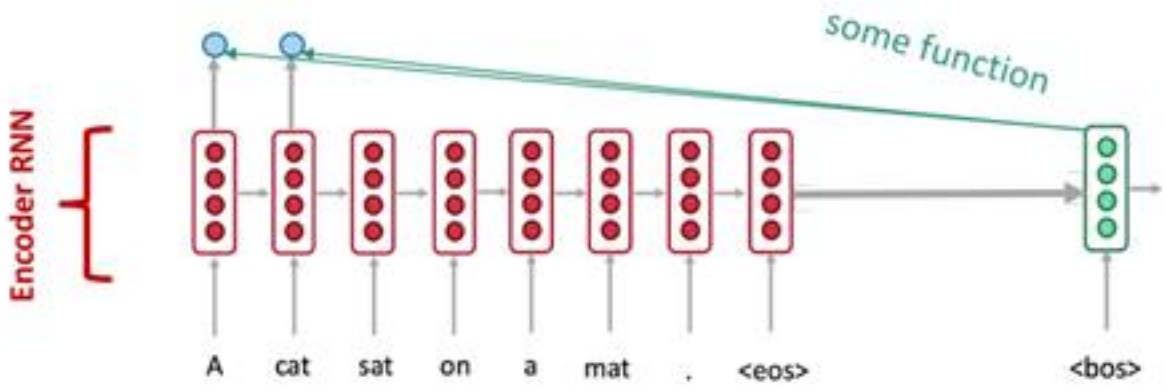
Attention



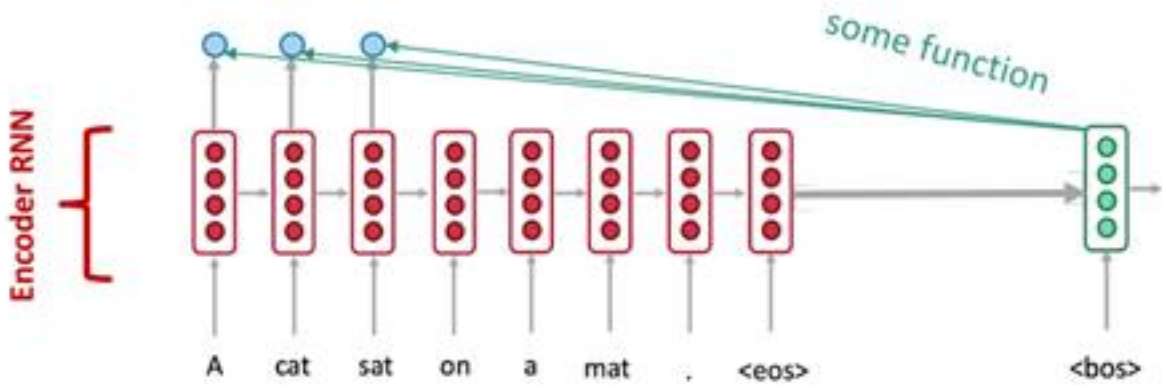
Attention



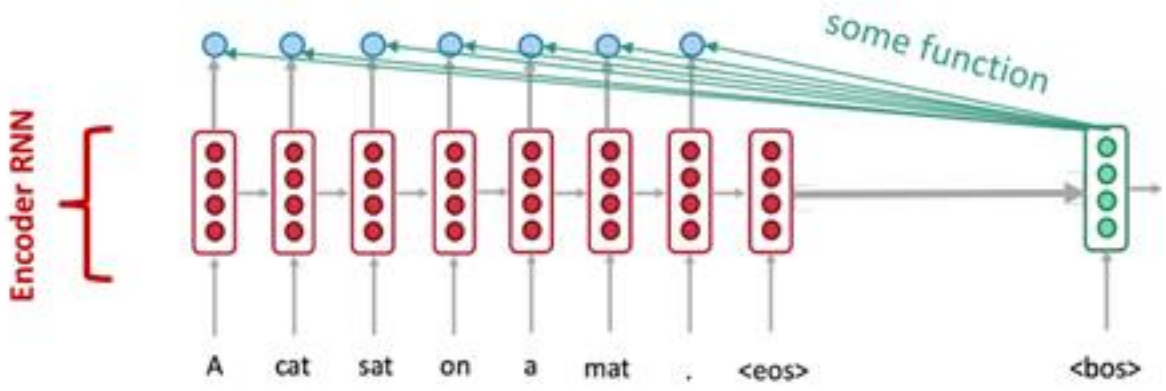
Attention



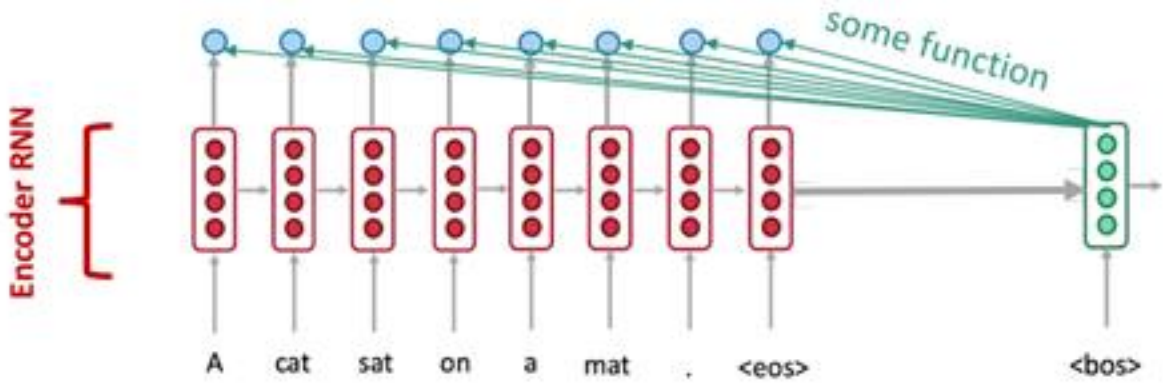
Attention



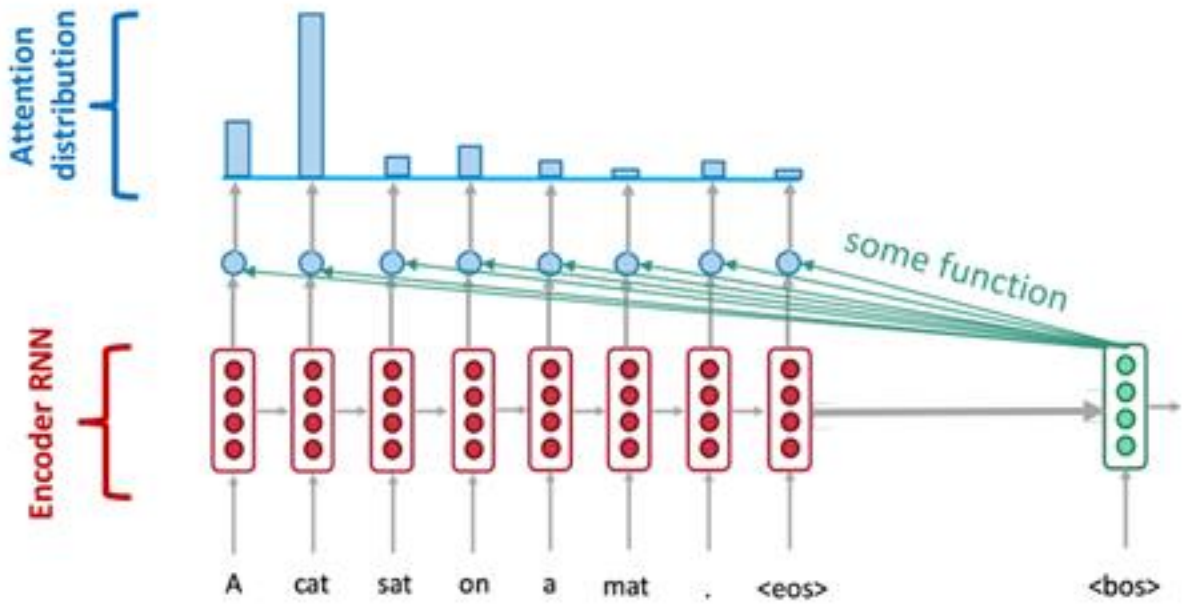
Attention



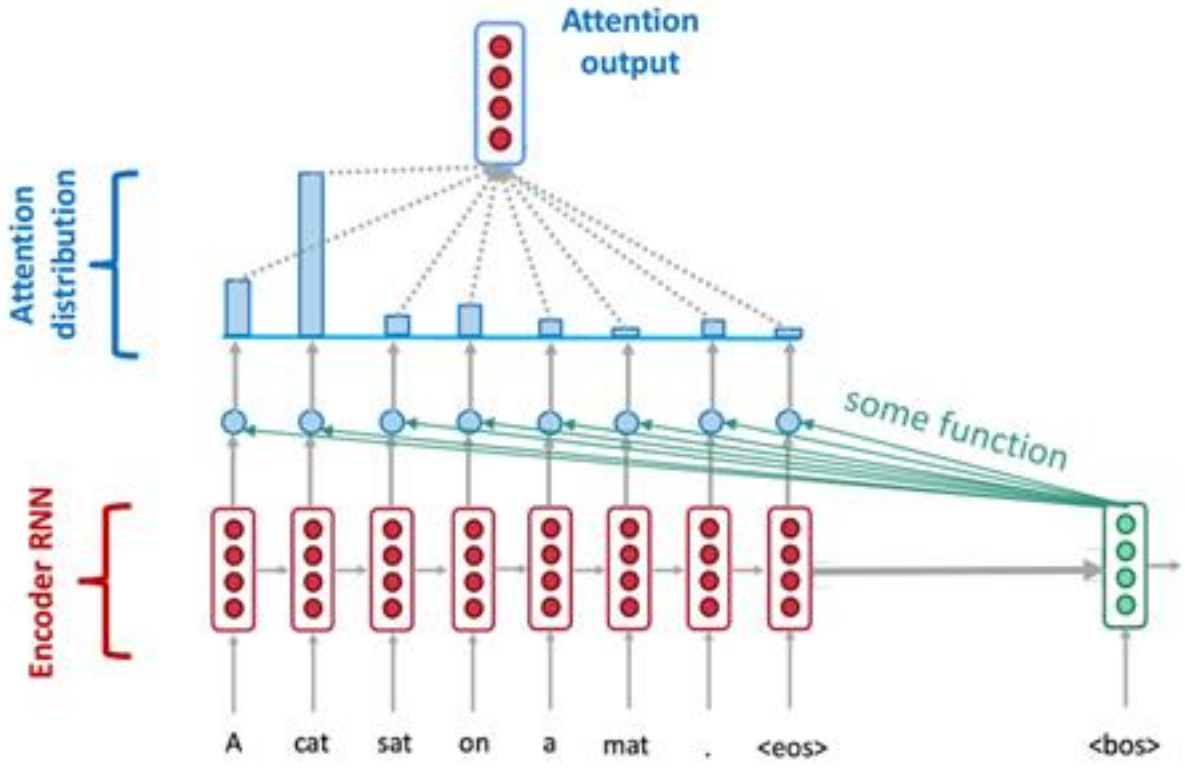
Attention



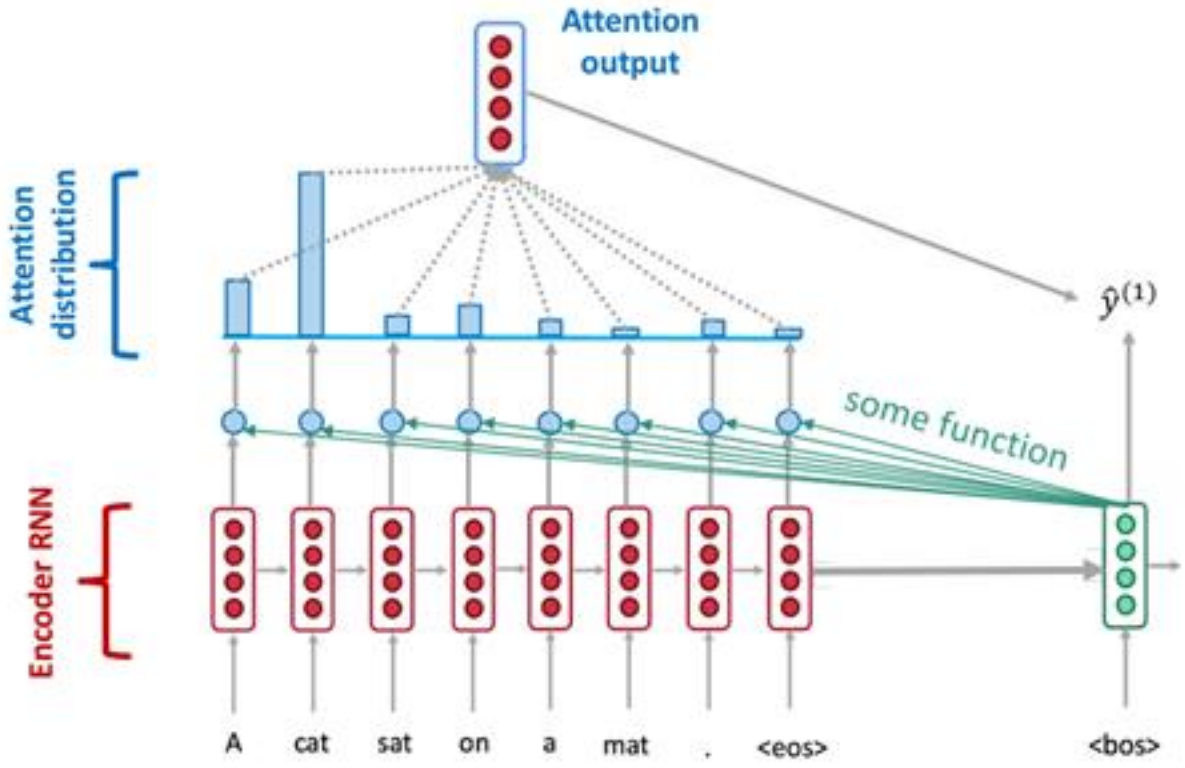
Attention


Attention distribution
Encoder RNN
some function
A
cat
sat
on
a
mat
.
<eos>
<bos>

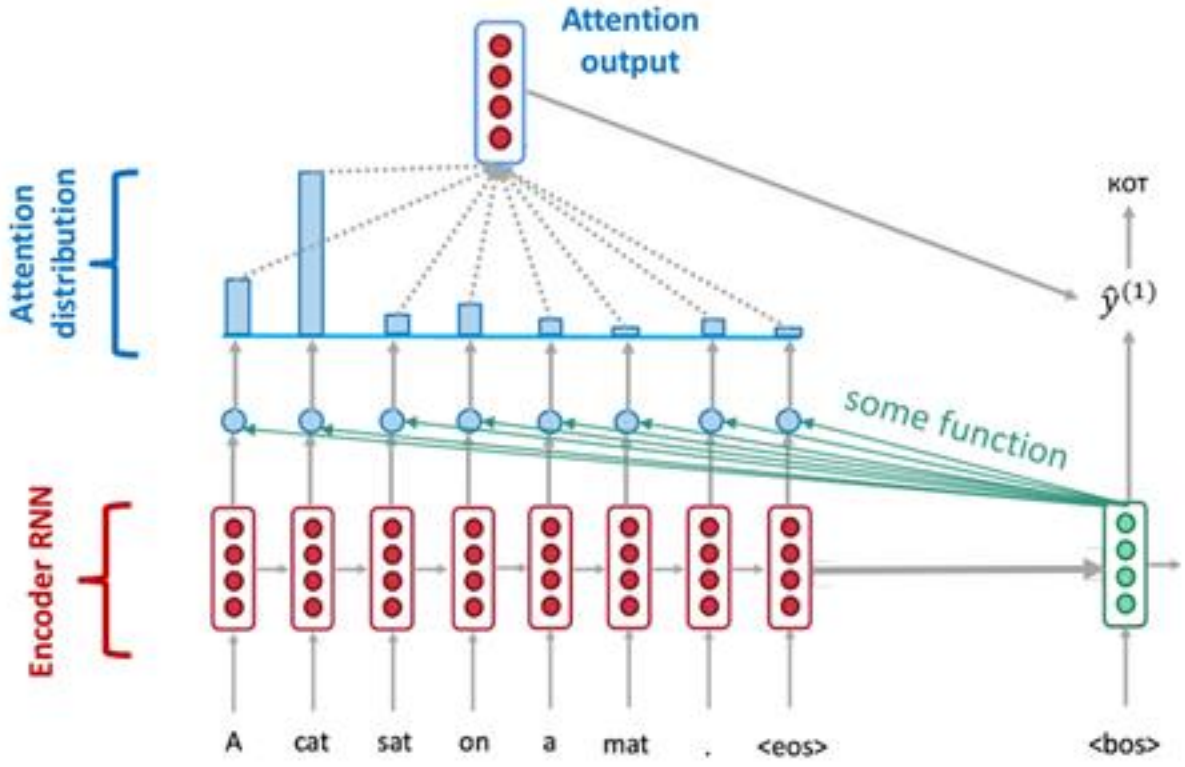
Attention


Attention output
Attention distribution
some function
Encoder RNN
A
cat
sat
on
a
mat
.
<eos>
<bos>

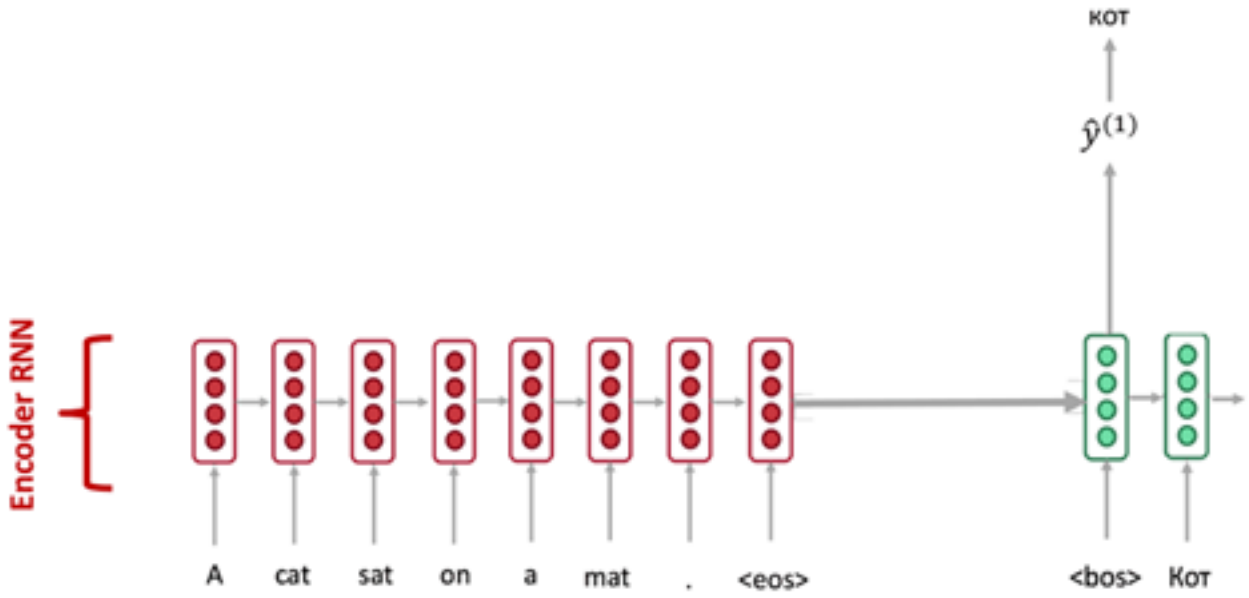
Attention


Attention output
Attention distribution
$\hat{y}^{(1)}$
some function
Encoder RNN
A
cat
sat
on
a
mat
.
<eos>
<bos>

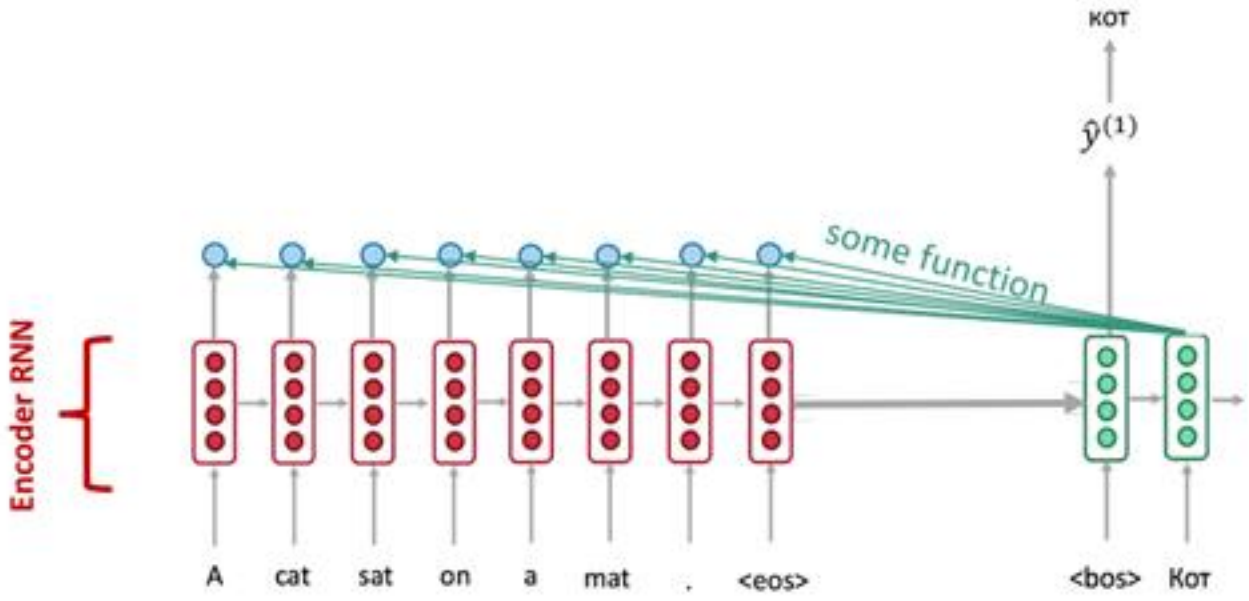
Attention


Attention output
Attention distribution
Encoder RNN
some function
KOT
$\hat{y}^{(1)}$
A
cat
sat
on
a
mat
.
<eos>
<bos>

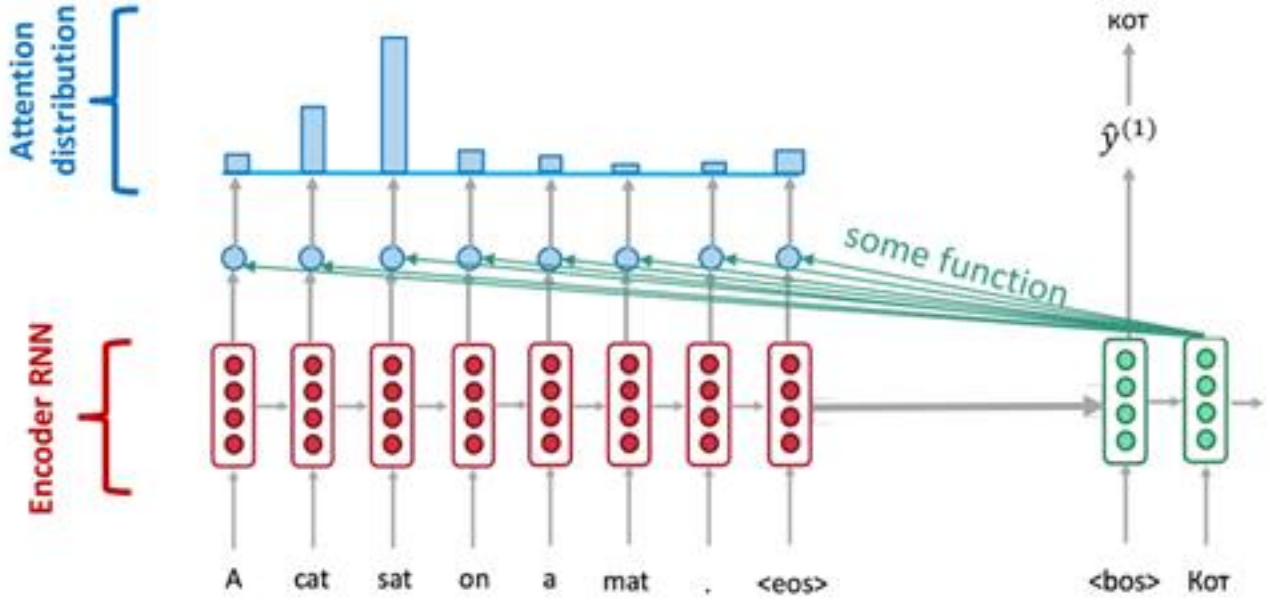
Attention



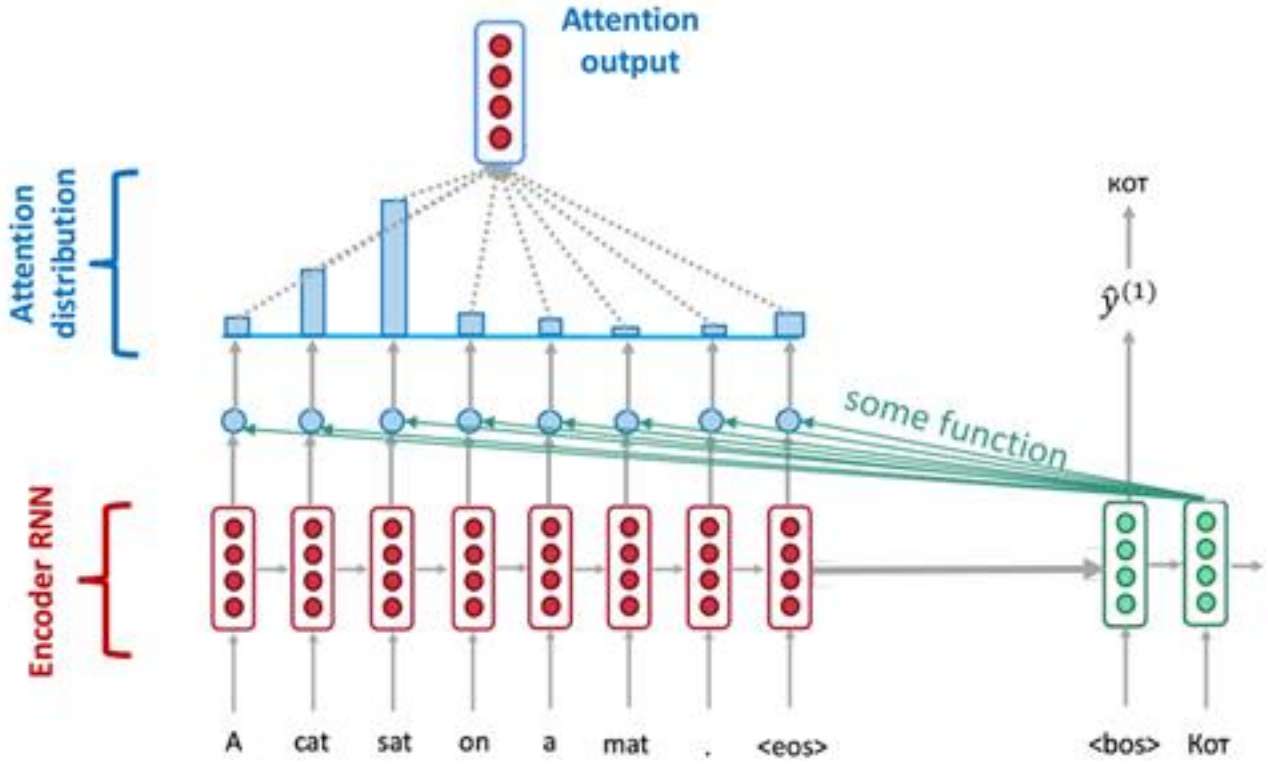
Attention


кот
$\hat{y}^{(1)}$
some function
Encoder RNN
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот

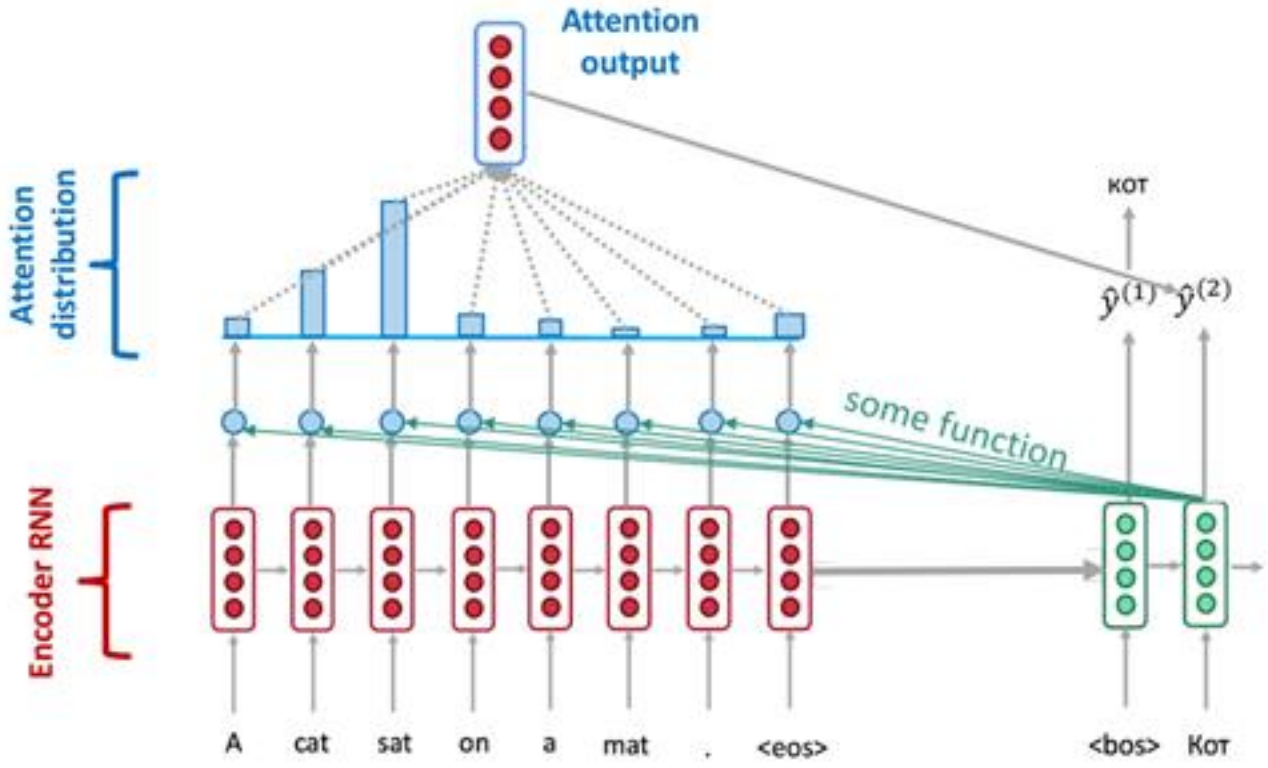
Attention


Attention distribution
Encoder RNN
some function
кот
$\hat{y}^{(1)}$
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот

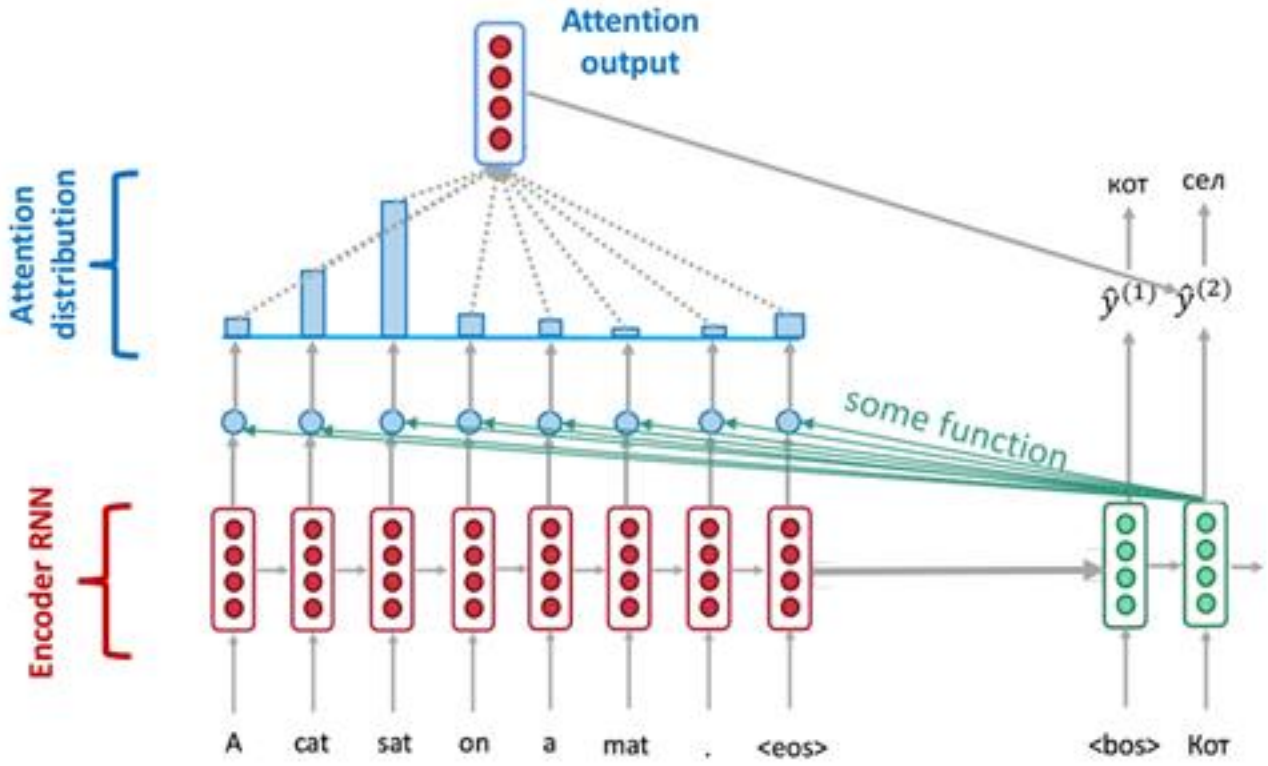
Attention


Attention output
Attention distribution
Encoder RNN
some function
кот
$\hat{y}^{(1)}$
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот

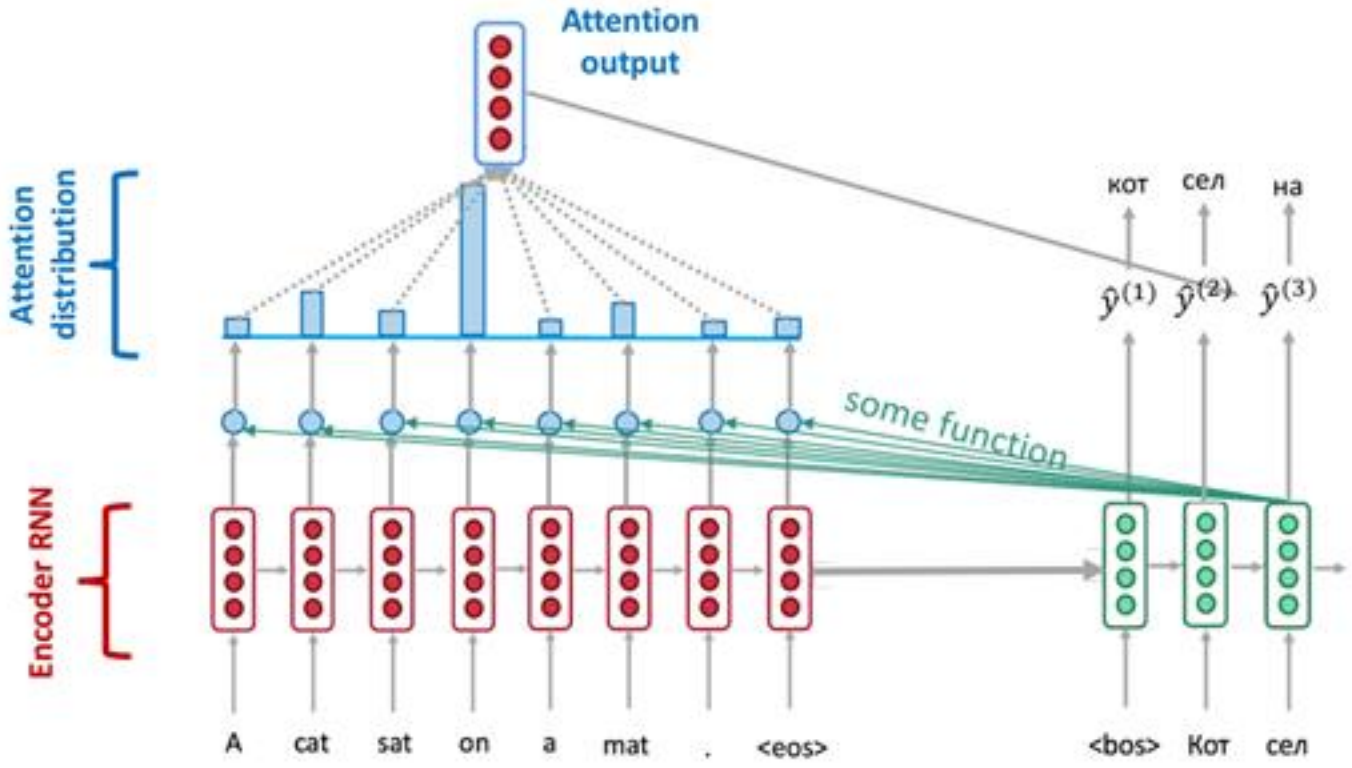
Attention


Attention output
Attention distribution
Encoder RNN
some function
кот
$\hat{y}^{(1)}$ $\hat{y}^{(2)}$
A cat sat on a mat . <eos>
<bos> Кот

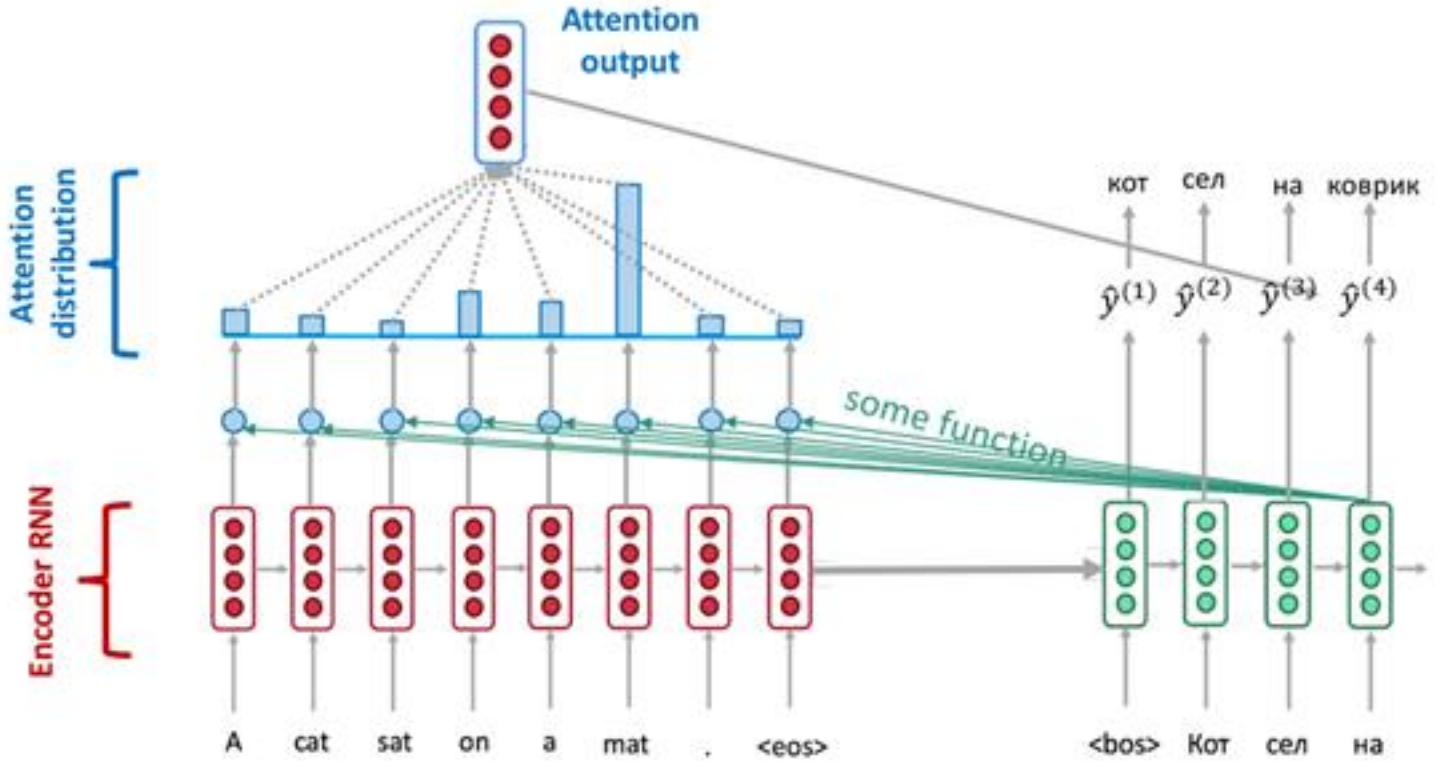
Attention


Attention output
Attention distribution
Encoder RNN
some function
кот
сел
$\hat{y}^{(1)}$
$\hat{y}^{(2)}$
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот

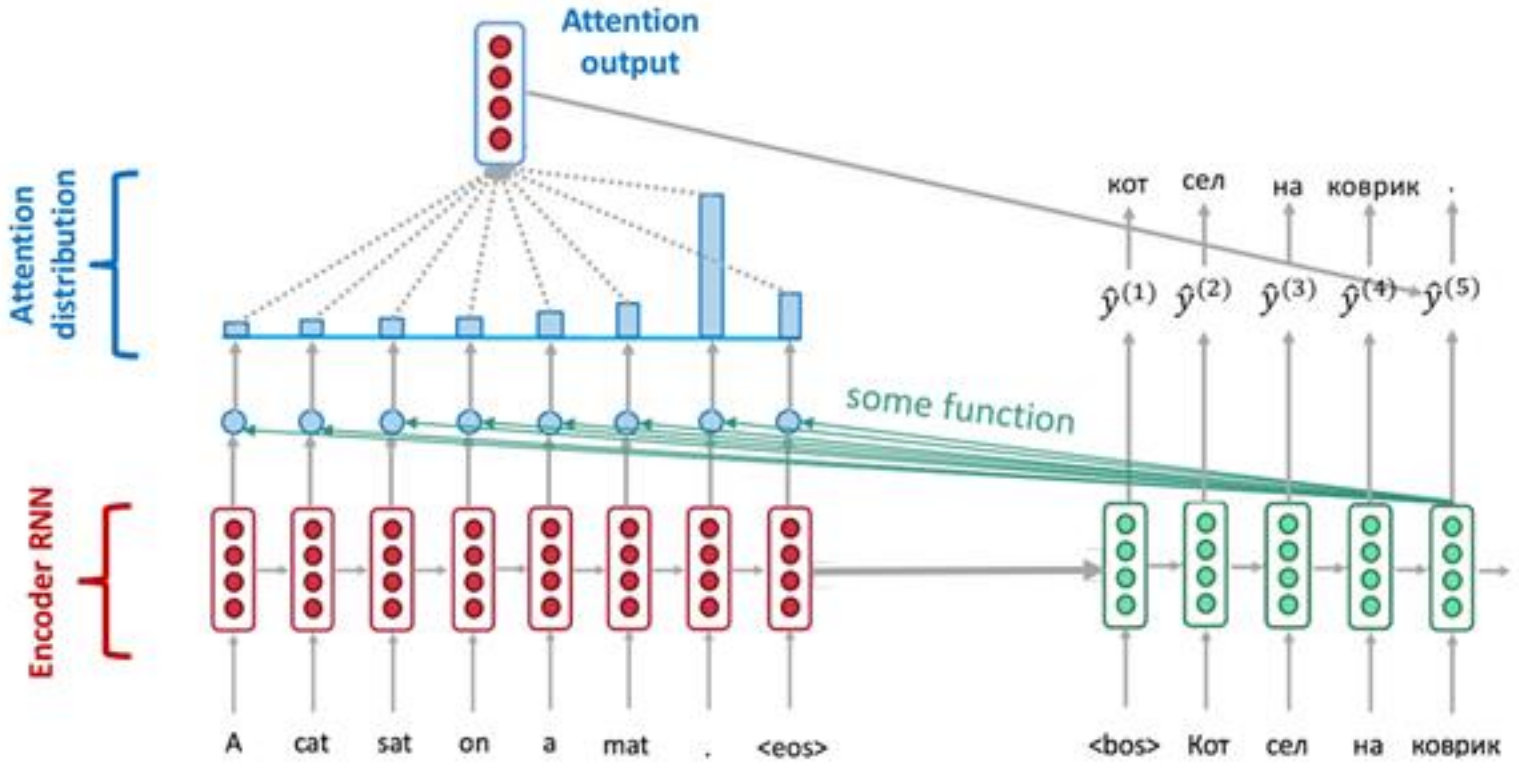
Attention


Attention output
Attention distribution
Encoder RNN
some function
кот
сел
на
$\hat{y}^{(1)}$
$\hat{y}^{(2)}$
$\hat{y}^{(3)}$
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот
сел

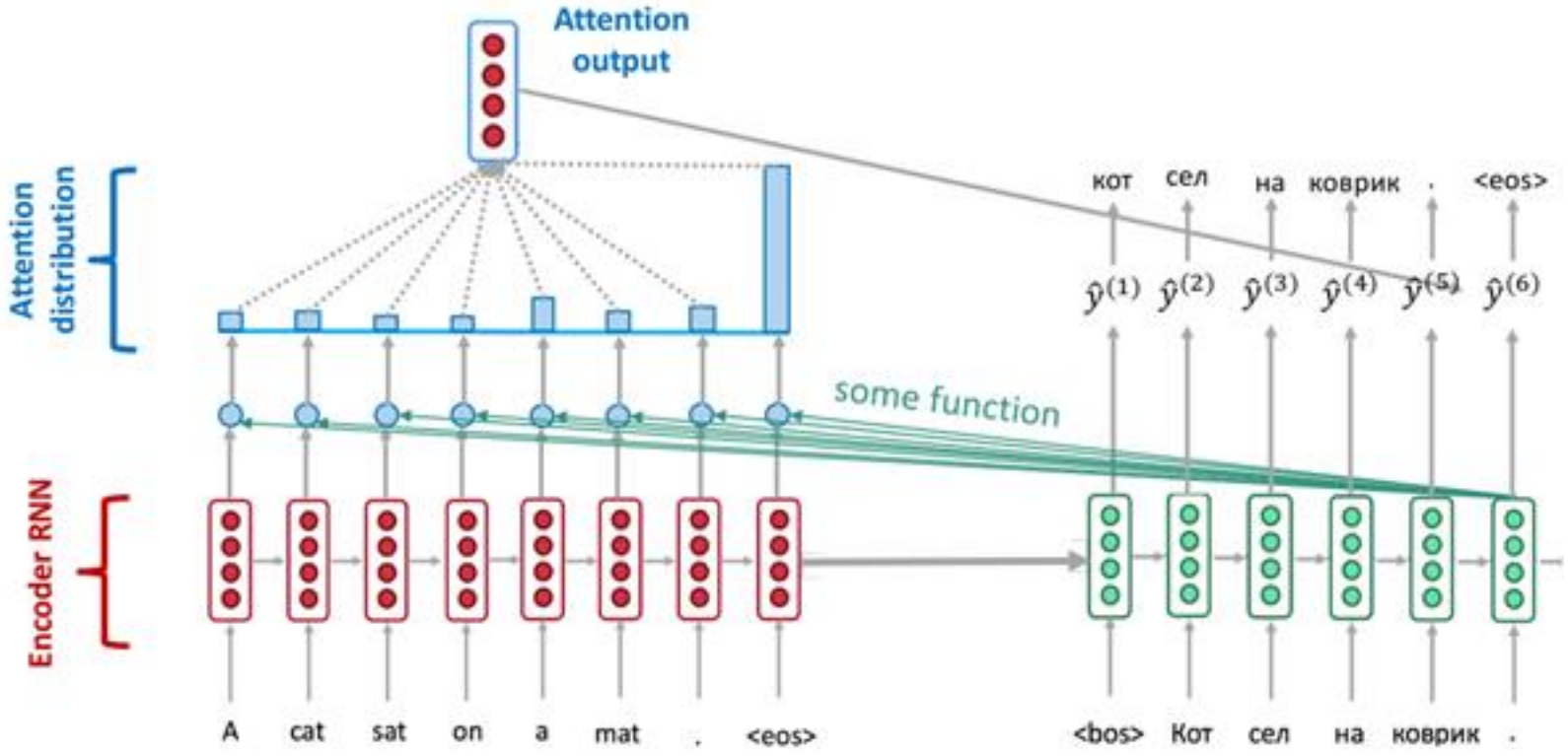
Attention


Attention output
Attention distribution
Encoder RNN
some function
кот
сел
на
коврик
$\hat{y}^{(1)}$
$\hat{y}^{(2)}$
$\hat{y}^{(3)}$
$\hat{y}^{(4)}$
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот
сел
на

Attention

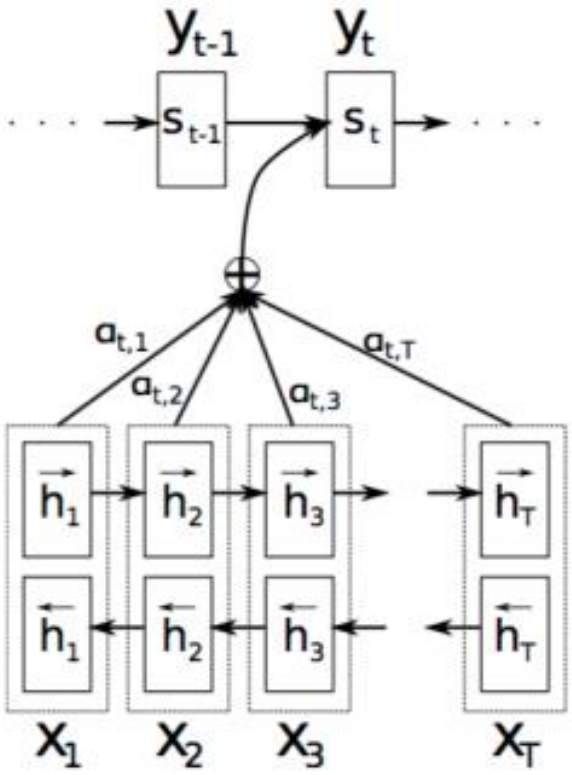

Attention output
Attention distribution
Encoder RNN
some function
A
cat
sat
on
a
mat
.
<eos>
<bos>
Кот
сел
на
коврик
кот
сел
на
коврик
.
$\hat{y}^{(1)}$
$\hat{y}^{(2)}$
$\hat{y}^{(3)}$
$\hat{y}^{(4)}$
$\hat{y}^{(5)}$

Attention


Attention output
Attention distribution
Encoder RNN
some function
A cat sat on a mat . <eos>
<bos> Кот сел на коврик .
кот сел на коврик . <eos>

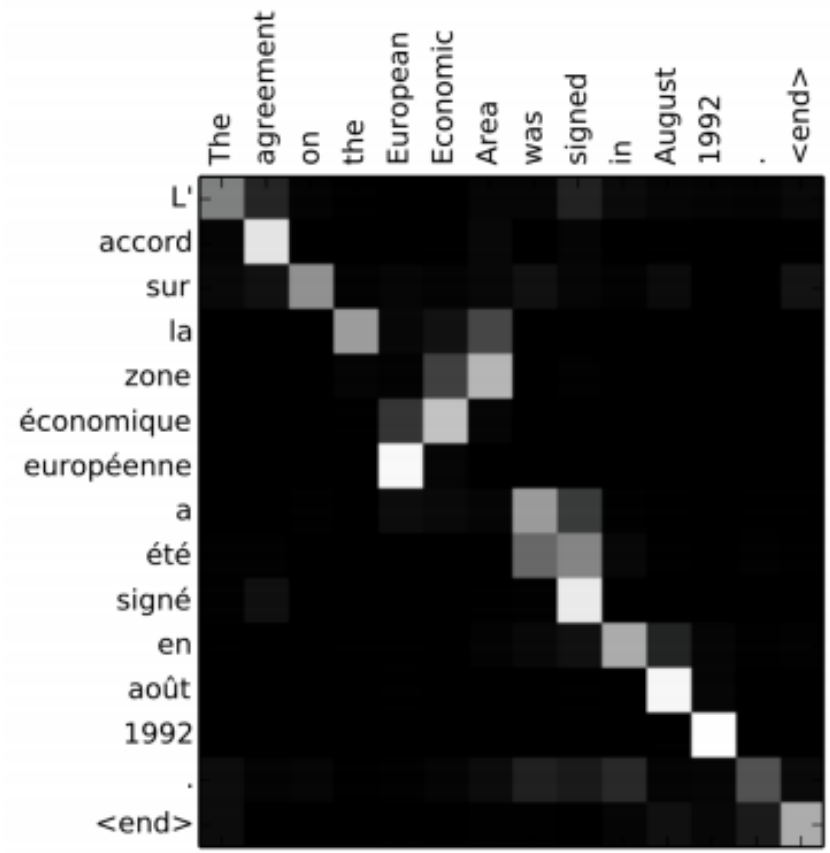
Neural Machine Translation by Jointly Learn to Align and Translate

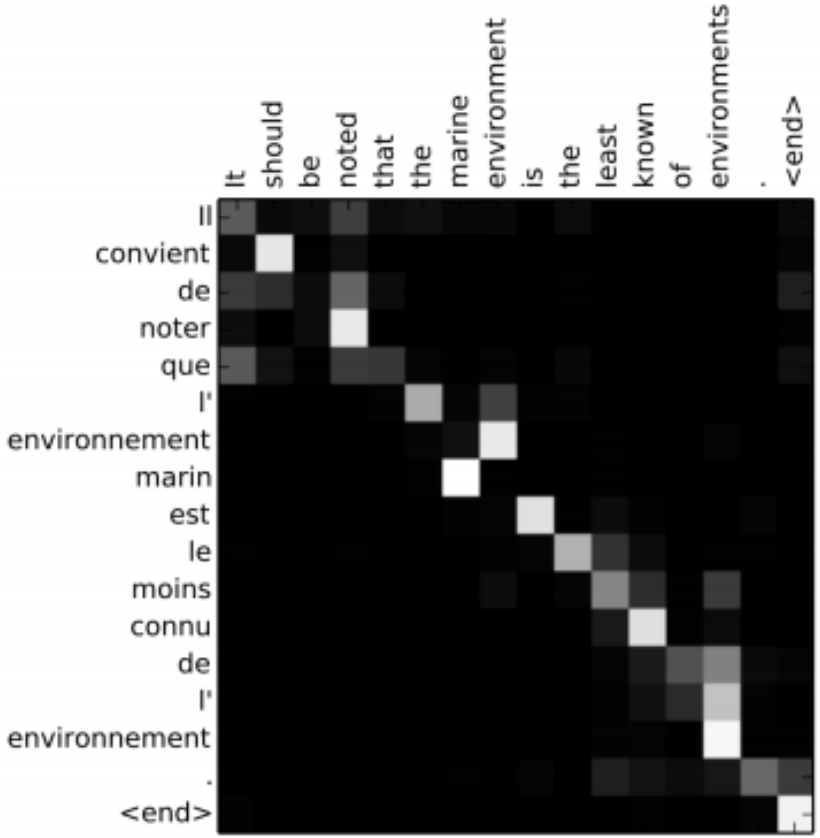
- **Преимущества внимания:**
  - Не нужно помнить все предложение
  - Нет фиксированного размера представления
  - «Похоже на людей»



1. Bahdanau et al, ICLR 2015, https://arxiv.org/pdf/1409.0473.pdf

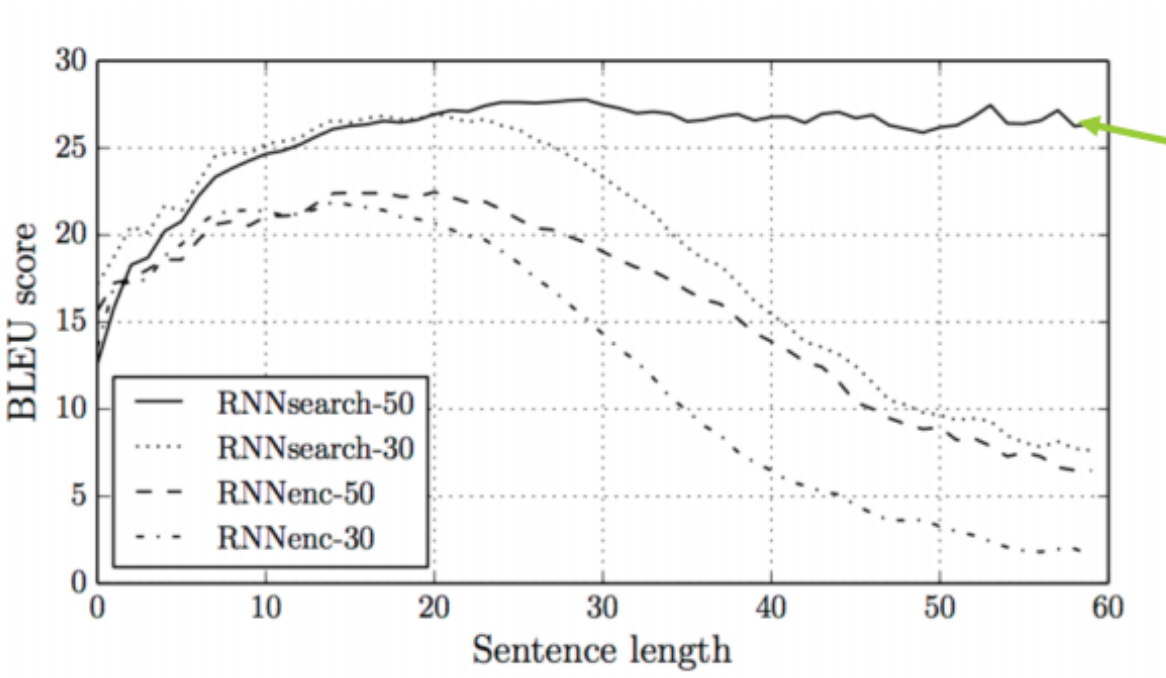
Neural Machine Translation by Jointly Learn to Align and Translate





1. Bahdanau et al, ICLR 2015, https://arxiv.org/pdf/1409.0473.pdf

Neural Machine Translation by Jointly Learn to Align and Translate



Нет потери качества для длинных педложений

1. Bahdanau et al, ICLR 2015, https://arxiv.org/pdf/1409.0473.pdf

Как вычислять Attention?

- dot product: $f_{att}(h_i, s_j) = h_i^T s_j$
- bilinear attention: $f_{att}(h_i, s_j) = h_i^T W_a s_j$
- multi-layer perceptron: $f_{att}(h_i, s_j) = v_a \tanh(W_a[s_j; h_i^T])$
- любая другая функция, которую можете себе представить ☺:

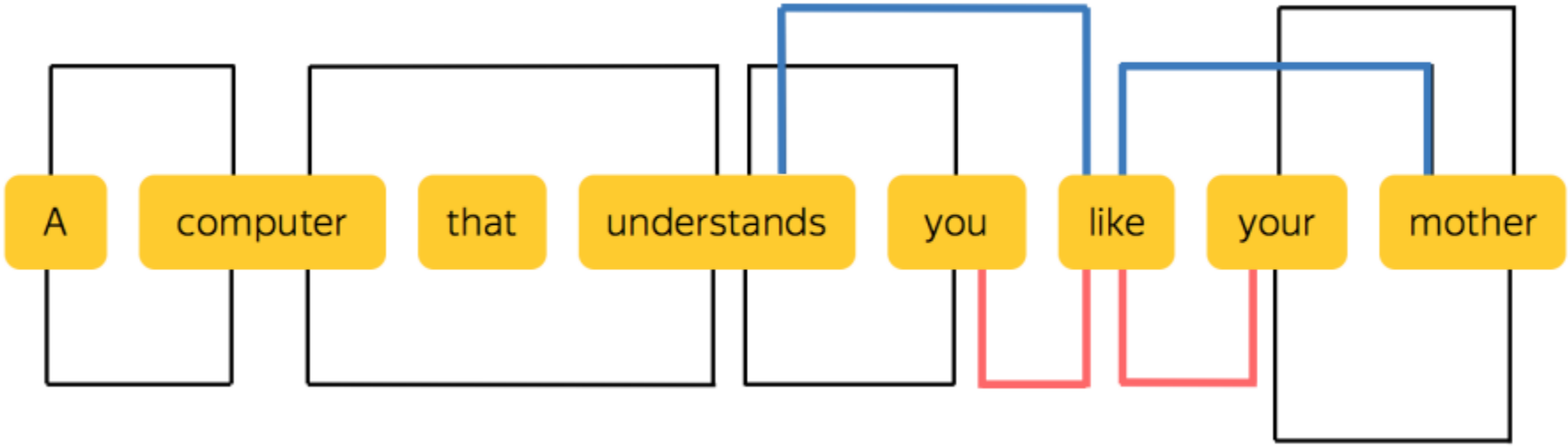
$$z(c, m, q) = \left[c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m\right]$$

$$G(c, m, q) = \sigma\left(W^{(2)} \tanh\left(W^{(1)} z(c, m, q) + b^{(1)}\right) + b^{(2)}\right)$$

A bit crazy, huh?

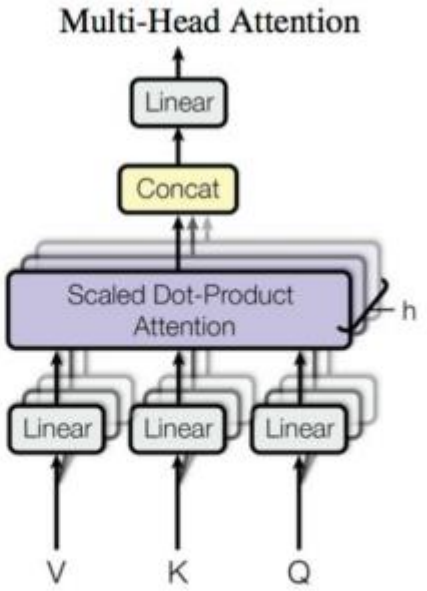
Transformer:
Attention is all you need!

Self Attention



Multi-Head Attention

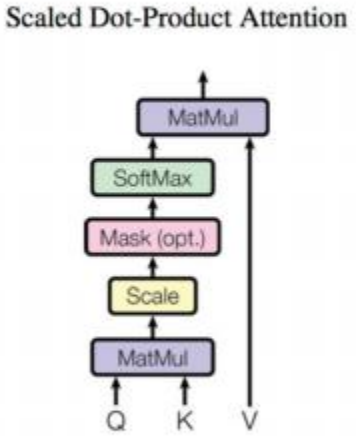
- Gender agreement
- Case government
- Lexical preferences
- ...

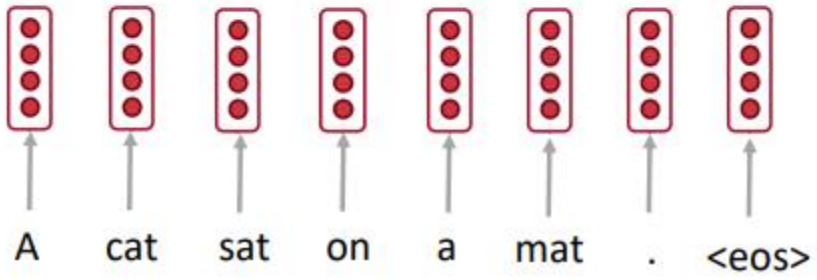


$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Self Attention

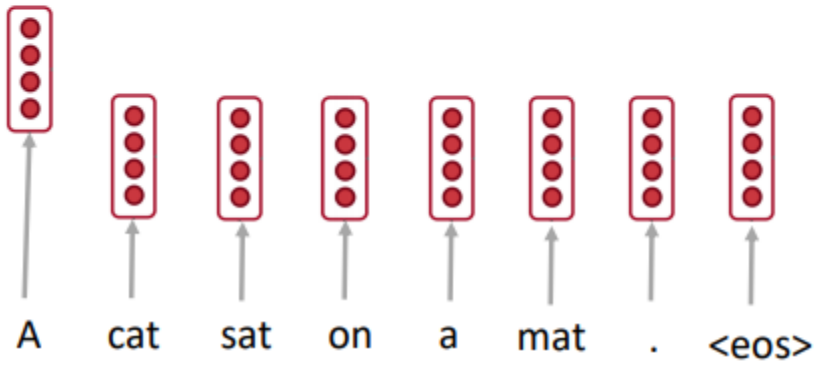
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

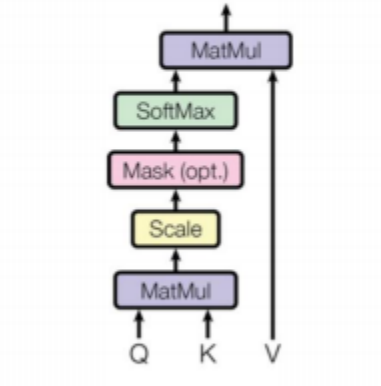




Self Attention

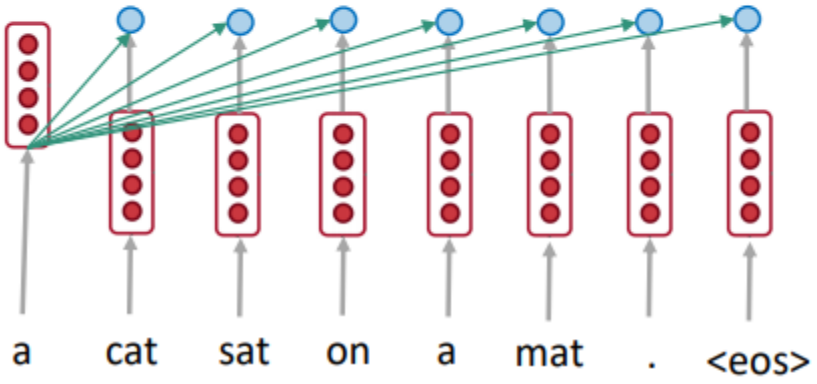
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

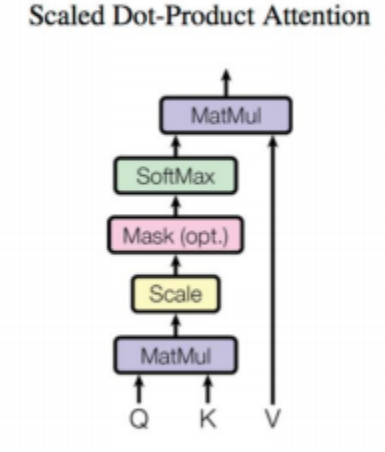




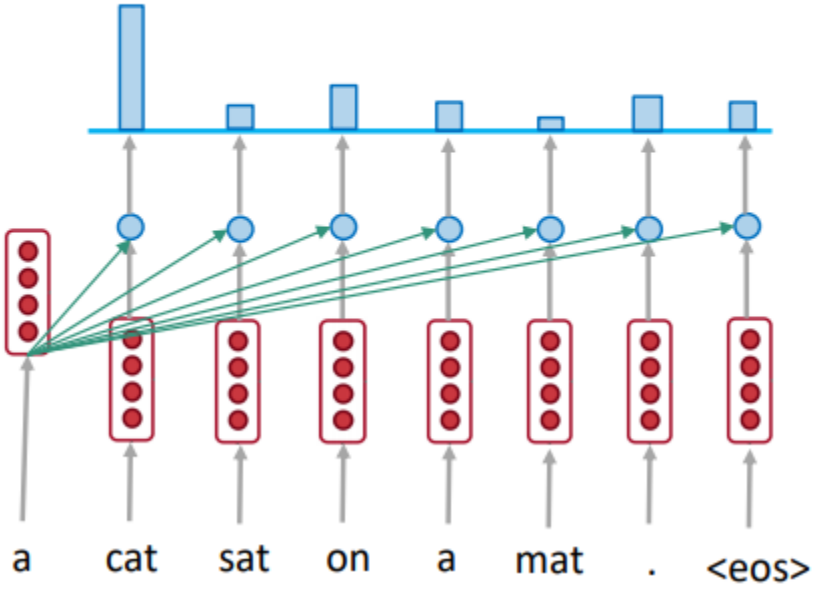
Self Attention

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

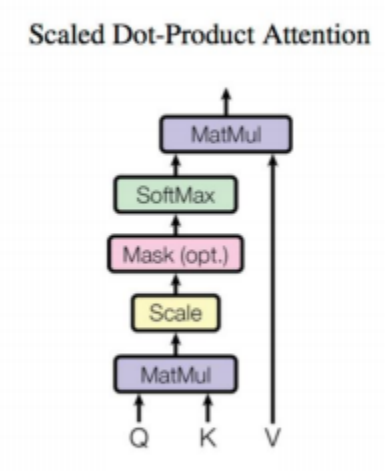




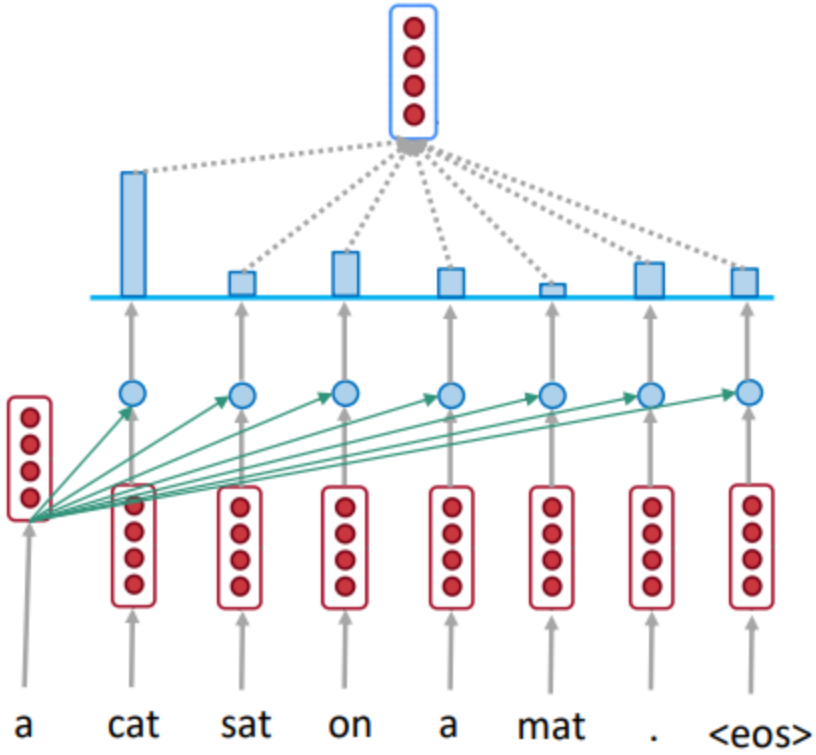
Self Attention



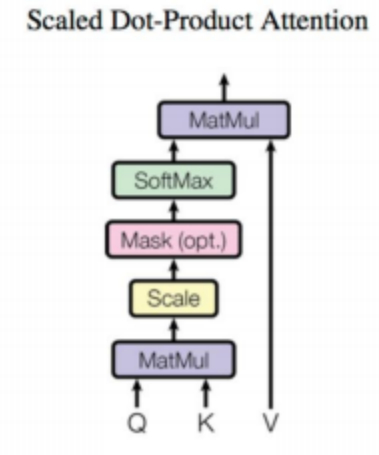
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Self Attention

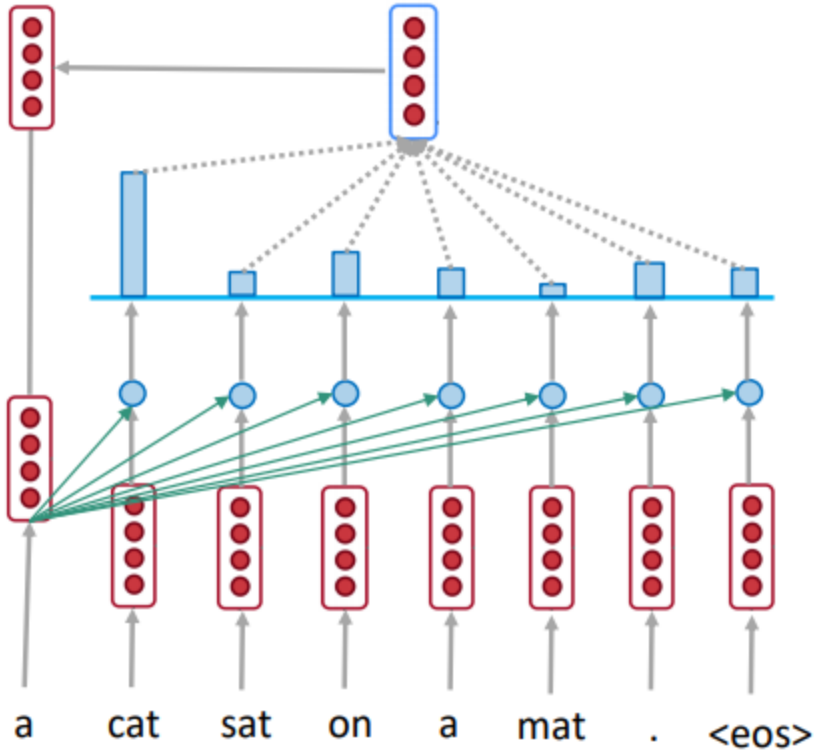


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

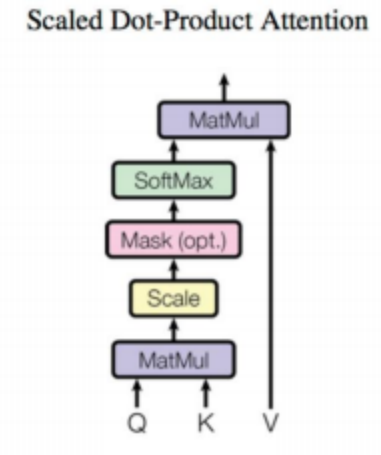


Self Attention

- Обновляем представление слова «a»

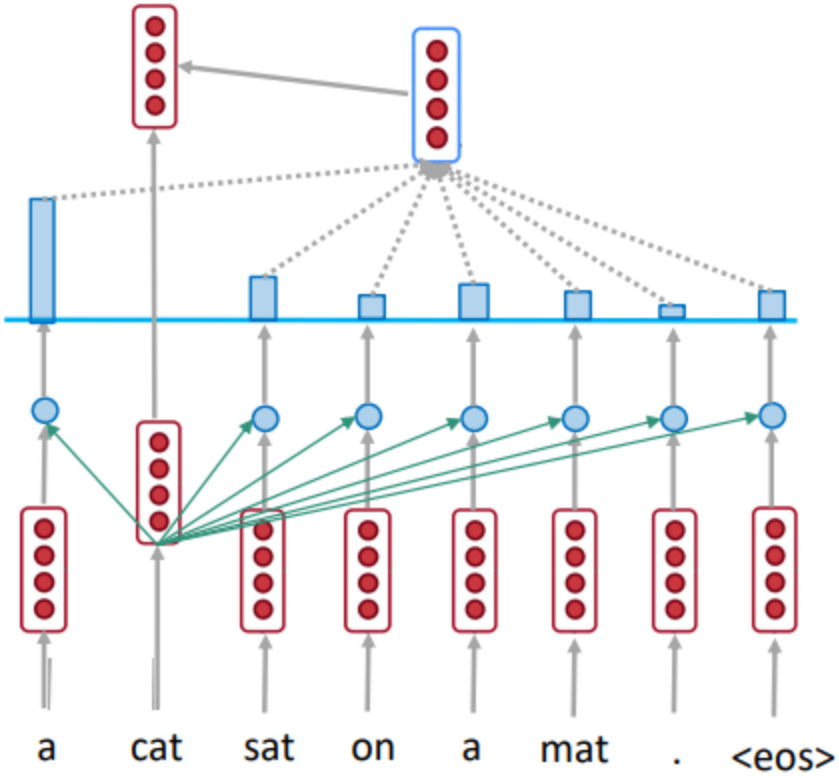


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

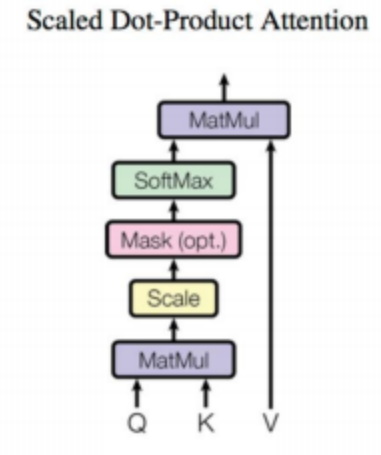


Self Attention

- Обновляем представление слова «cat»

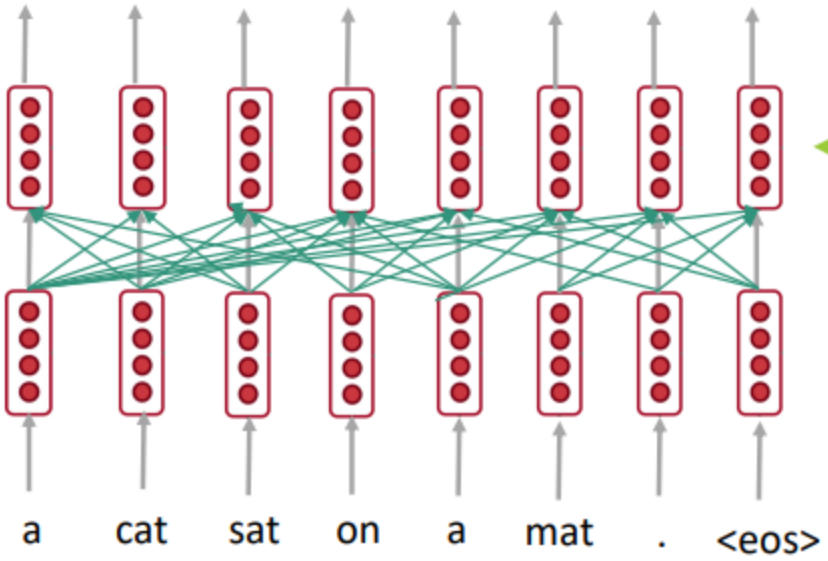


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



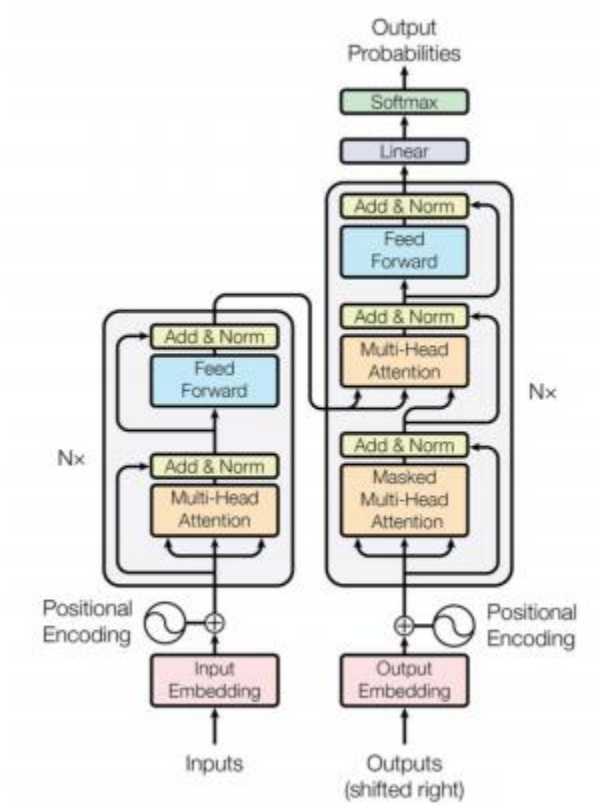
Self Attention

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

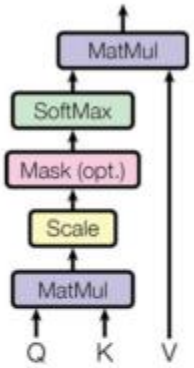


Updated representations for each word (one layer)

Transformer

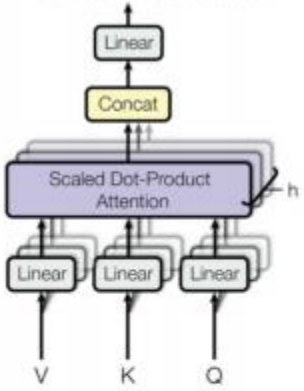


Scaled Dot-Product Attention



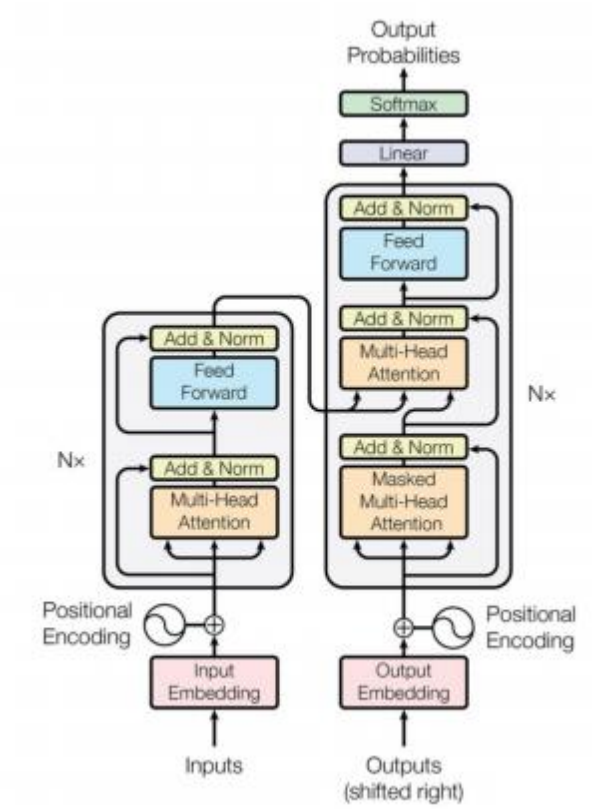
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Multi-Head Attention



1. Vaswani et al, NIPS 2017, https://papers.nips.cc/paper/7181-attention-is-all-you-need

Transformer



- Преимущества:
  - Нет рекуррентности, параллельное кодирование
  - Быстрое обучение: encoder и decoder могут быть параллельны
  - Нет длинных связей: O(1) для любых токенов
  - Три внимания: модель не помнит слишком много
  - Multi-head attention позволяет обращать внимание на разные аспекты

1. Vaswani et al, NIPS 2017, https://papers.nips.cc/paper/7181-attention-is-all-you-need

Transformer

1. Визуализация: https://mchromiak.github.io/articles/2017/Sep/12/Transformer-Attention-is-all-you-need/#.W_Z9MfkzaM8

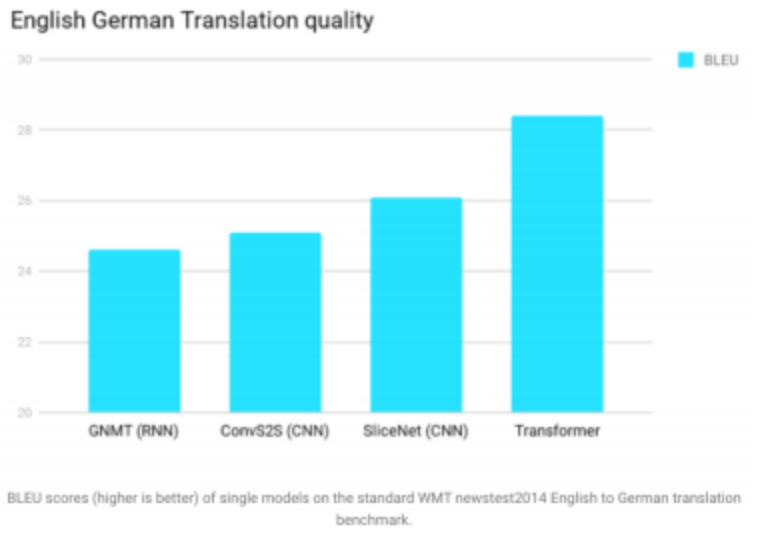
Transformer

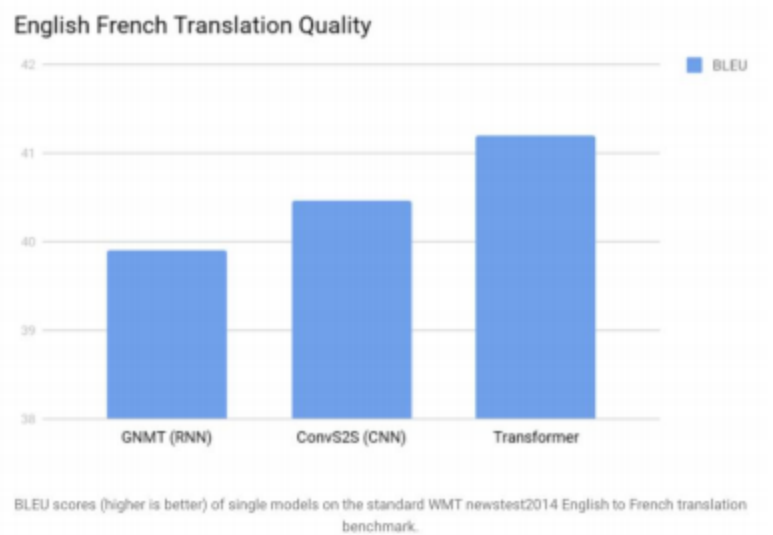
The animal didn't cross the street because it was too tired.
L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

The animal didn't cross the street because it was too wide.
L'animal n'a pas traversé la rue parce qu'elle était trop large.

1. Vaswani et al, NIPS 2017, https://papers.nips.cc/paper/7181-attention-is-all-you-need

Transformer

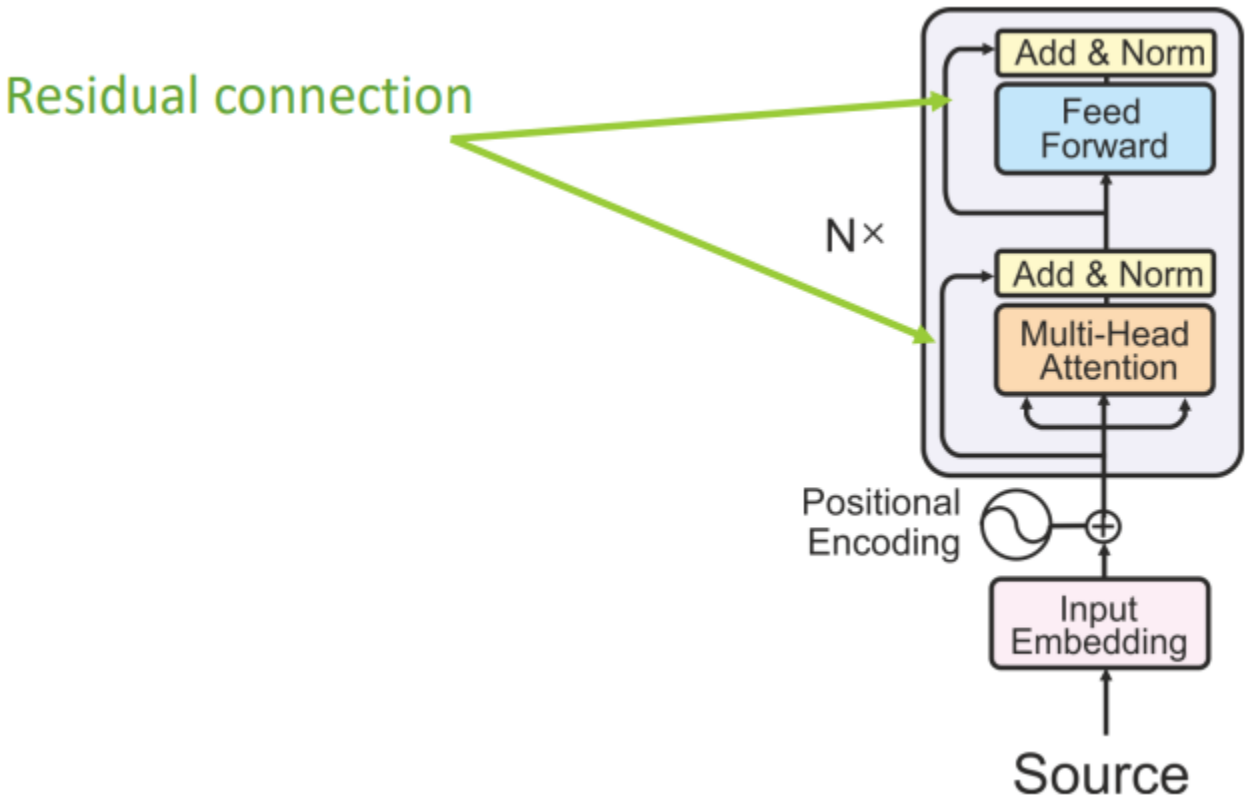




to French translation (one of eight attention heads).

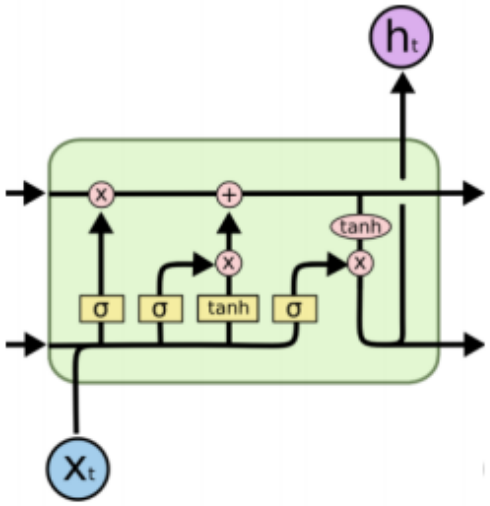
1. Vaswani et al, NIPS 2017, https://papers.nips.cc/paper/7181-attention-is-all-you-need

Почему residual connection?



Почему residual connection? Пример

- LSTM and long-term dependences



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Original version:

or

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

$$C_t = C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh\left(C_t\right)$$

- Почему проблема vanishing/exploding здесь не настолько существенна?

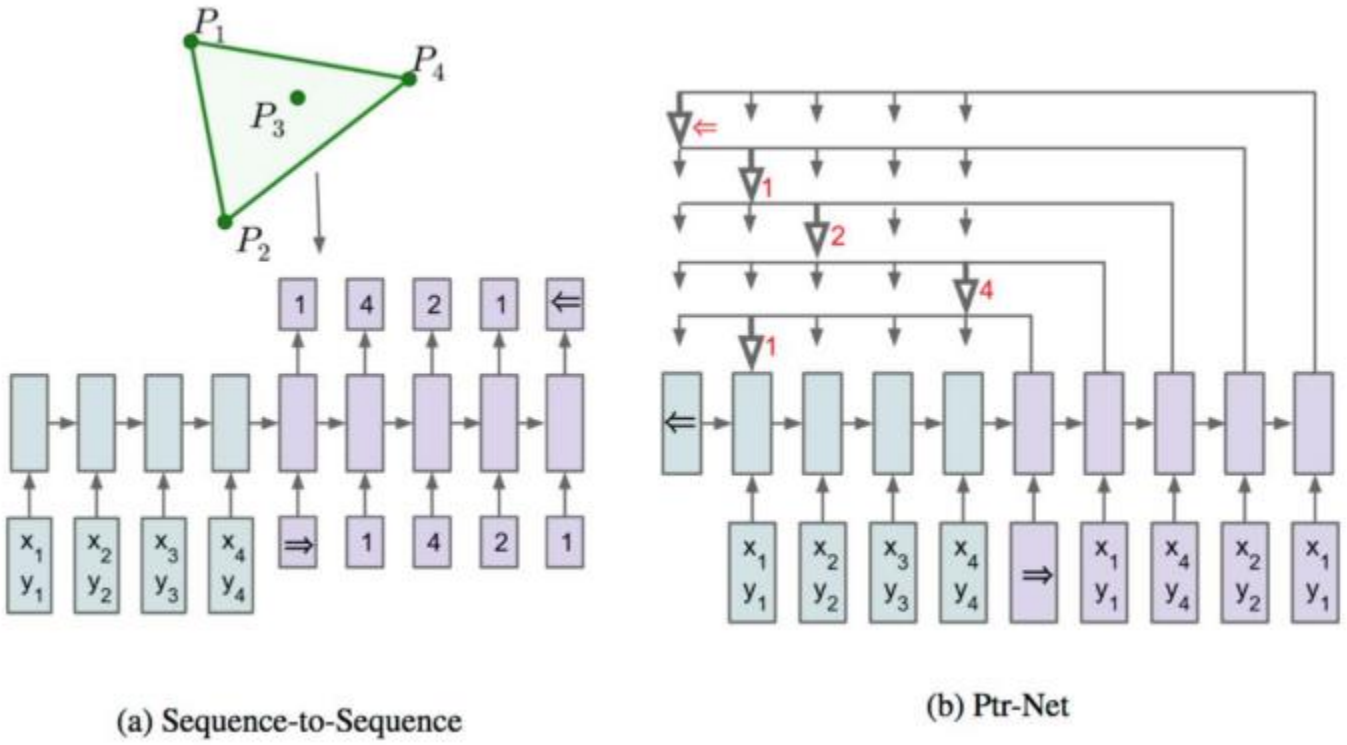
Что если мы не знаем размер словаря?

- Pointer Networks

Attention weights

=

output distribution



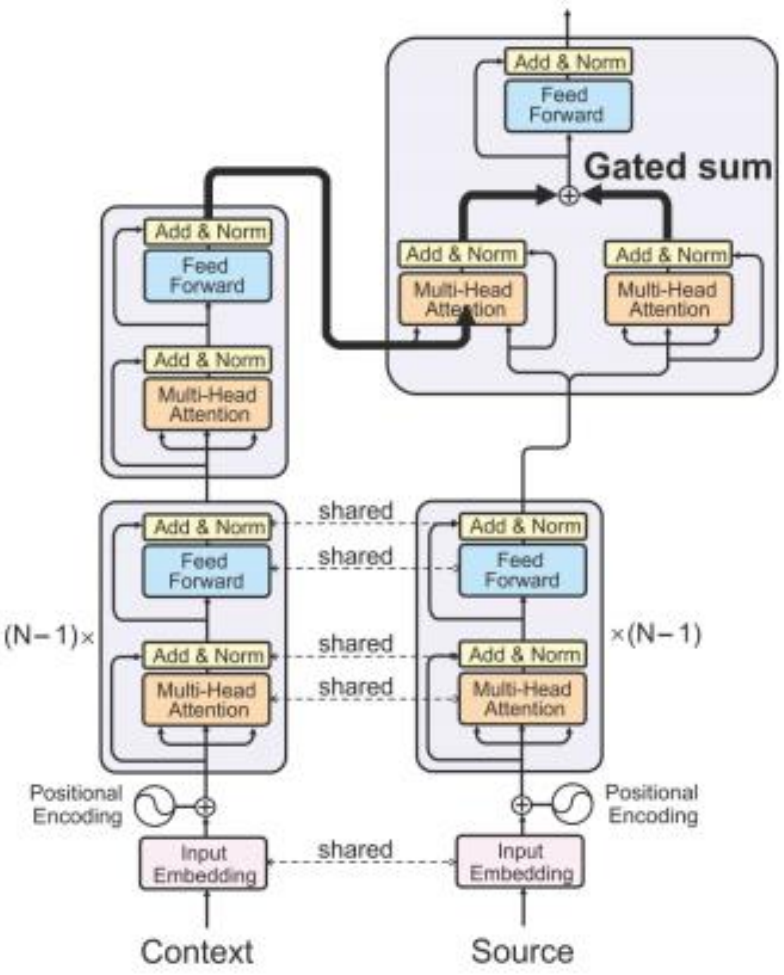
(a) Sequence-to-Sequence

(b) Ptr-Net

1. Vinyals et al, NIPS 2015, https://papers.nips.cc/paper/5866-pointer-networks.pdf

Что если ввести контекст в NMT систему?

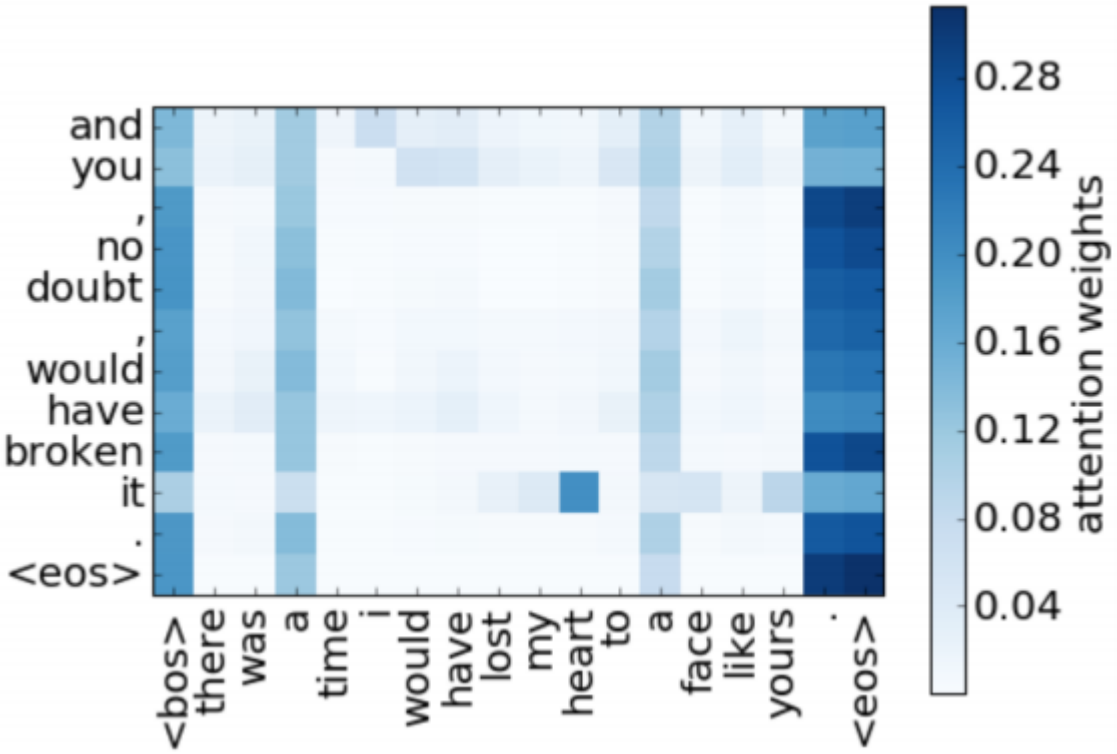
- Context-aware NMT для обнаружения анафоры
- Начинаем с Transformer [Vaswani et al, 2018]
- Включаем информацию о context на стороне encoder
- Используем разделение encoder и context
- Используем первые N-1 слоев source и context
- Последний слой включает в себя контекстную информацию



1. Voita et al, ACL 2018, http://www.aclweb.org/anthology/P18-1117

Что если ввести контекст в NMT систему?

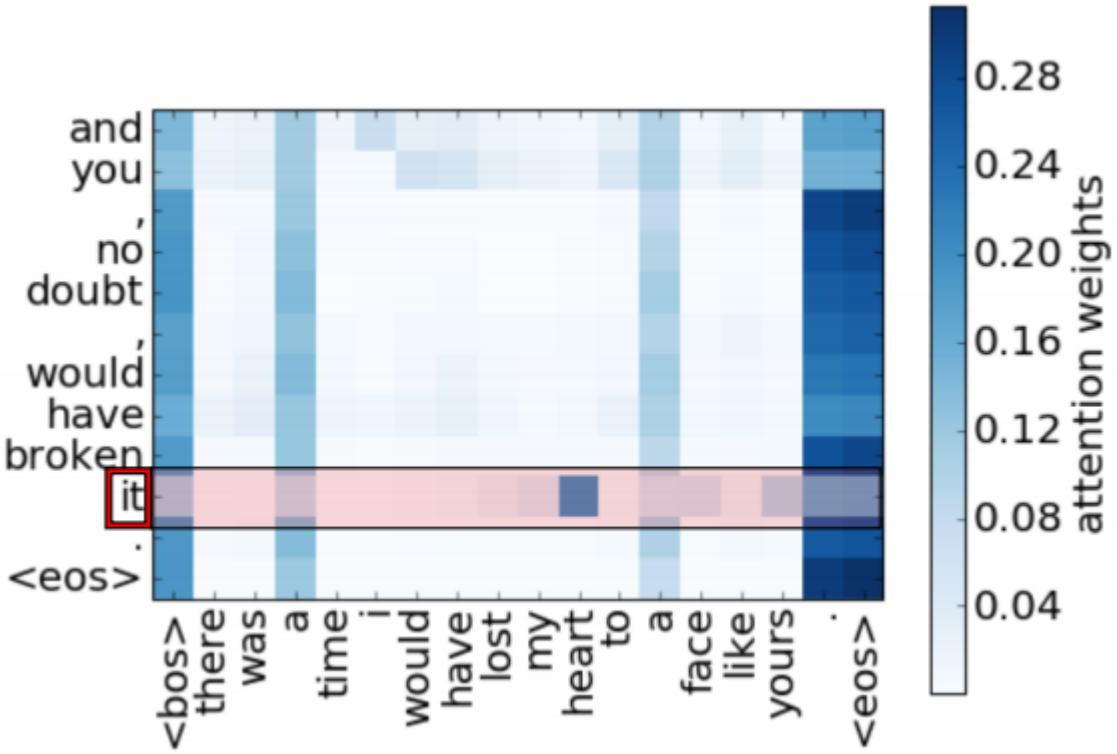
- Context:
  - There was a time I would have lost my heart to a face like yours.
- Source:
  - And you, no doubt, would have broken **it**.



1. Voita et al, ACL 2018, http://www.aclweb.org/anthology/P18-1117

Что если ввести контекст в NMT систему?

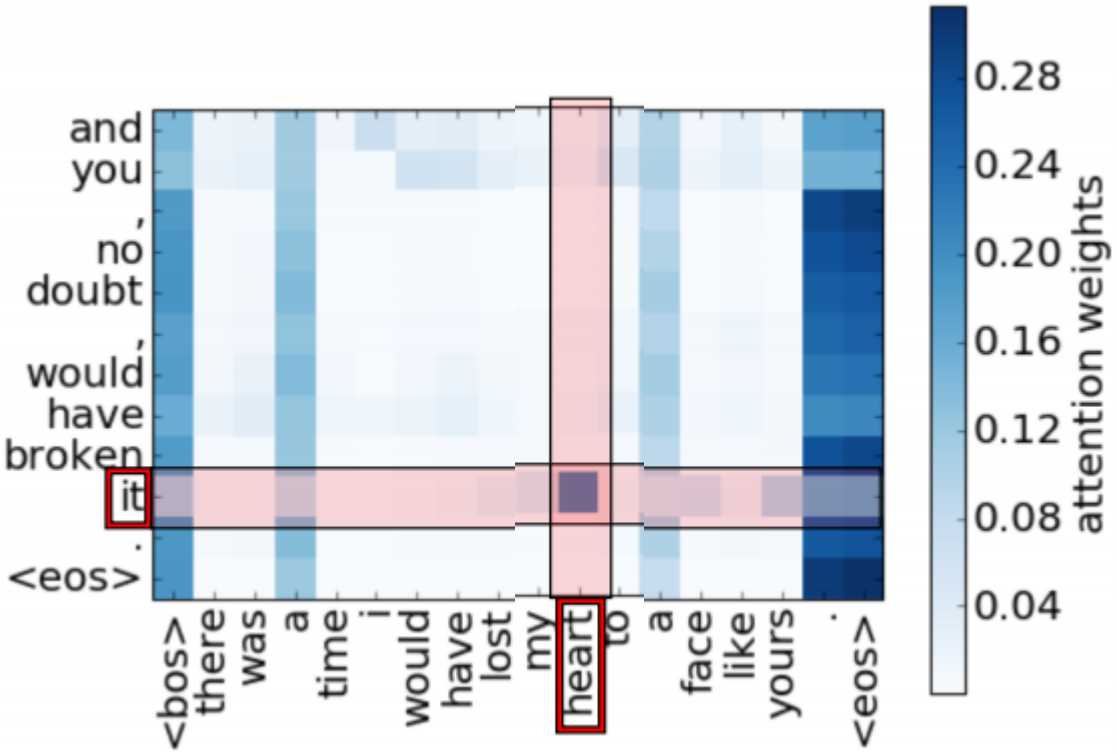
- Context:
  - There was a time I would have lost my heart to a face like yours.
- Source:
  - And you, no doubt, would have broken **it**.



1. Voita et al, ACL 2018, http://www.aclweb.org/anthology/P18-1117

Что если ввести контекст в NMT систему?

- Context:
  - There was a time I would have lost my heart to a face like yours.
- Source:
  - And you, no doubt, would have broken **it**.



1. Voita et al, ACL 2018, http://www.aclweb.org/anthology/P18-1117

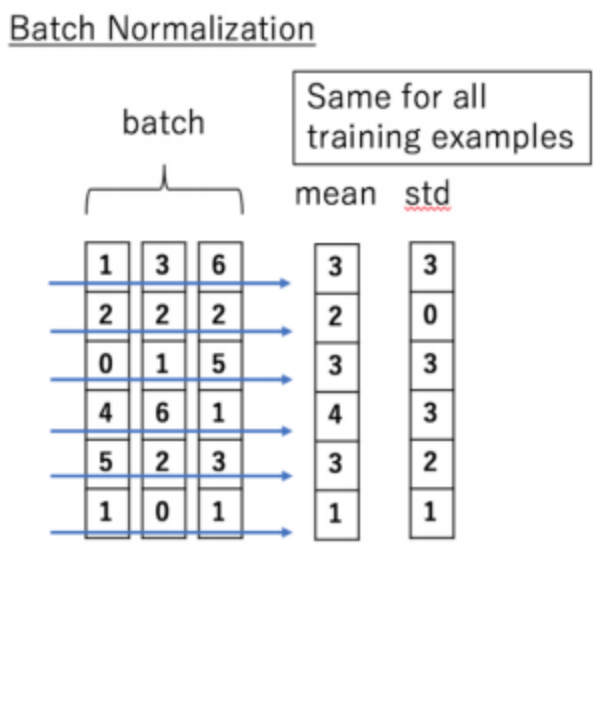
Layer norm

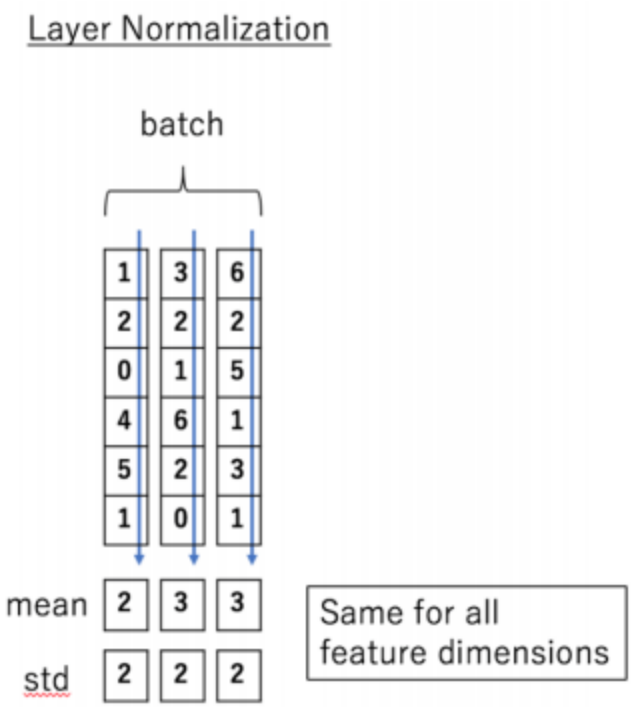
Batch normalization

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}$$
$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_{ij} - \mu_j)^2$$
$$\hat{x_{ij}} = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}$$

Layer normalization:

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_{ij}$$
$$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m}(x_{ij} - \mu_i)^2$$
$$\hat{x_{ij}} = \frac{x_{ij} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$





1. Paper: Ba et al, 2016, https://arxiv.org/abs/1607.06450
2. Pictures: http://mlexplained.com/2018/01/13/weight-normalization-and-layer-normalization-explained-normalization-in-deep-learning-part-2/

Применения Attention

- Classification
- Natural Language Inference
- NER
- Question Answering
- Sentence parse trees
- Image captioning
- …

СПАСИБО ЗА ВНИМАНИЕ

https://github.com/king-menin/nlp-hse-autumn2018

Литература

- NLP курс в яндексе https://github.com/yandexdataschool/nlp_course
- Курс в Stanford CS224N http://cs224n.stanford.edu/
- CMU lectures seq2seq and attention
- BLEU and CIDEr
- Image captioning
  - MSCOCO captioning challenge
  - Captioning baseline notebook
- Lecture on attention mechanisms - video (RUSSIAN)
- Illustrated transformer post