

# Глубокое обучение и вообще

Кирпа Вадим

25 января 2023 г.

**Посиделка 8:** Attention и Transformer

# Agenda

- seq2seq
- Attention
- Self-attention
- Transformer
- ELMo, BERT
- Linformer

seq2seq

seq2seq

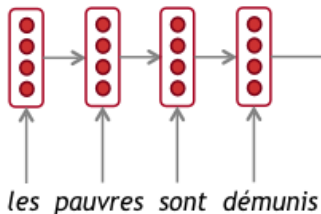
Самая стандартная подобная задача - машинный перевод. Нейронные сети ворвались в эту сферу человеческого прогресса в 2014 году

# BLEU (Bilingual evaluation understudy)

Если в кратце, то эта метрика сравнения полученного машиной перевода и человеческого, насколько мы вообще бьемся

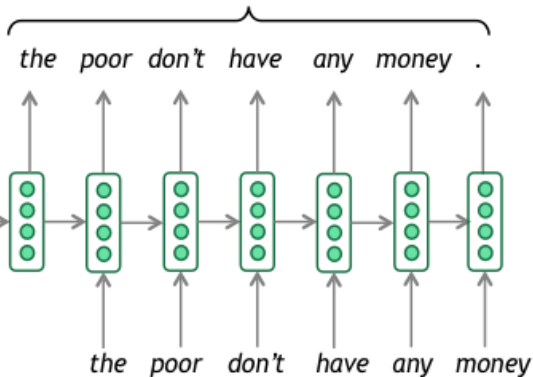
Из проблем данной метрики - если машина перевела правильно, но альтернативно, то BLEU будет низкий....

We feed in each word from left to right, one at a time. By the end, the NMT system has encoded information about the whole sentence in a numerical format.



French sentence (input)

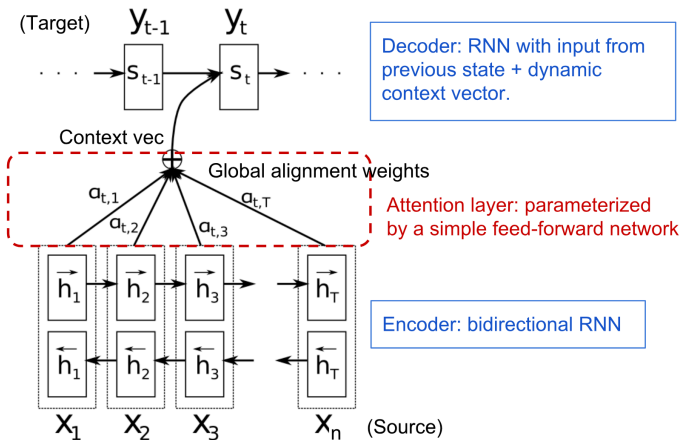
English translation (output)



The previous outputted word gets added as part of the input into the network next, giving the network some view of the sentence already produced and some context of the words preceding it.

Attention!

# Attention

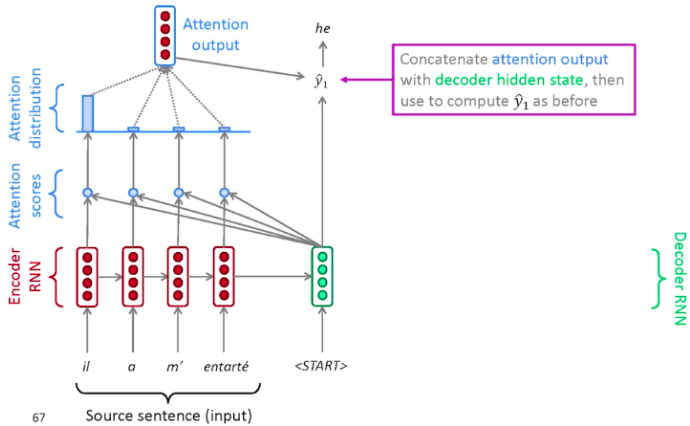


**Additive Attention**



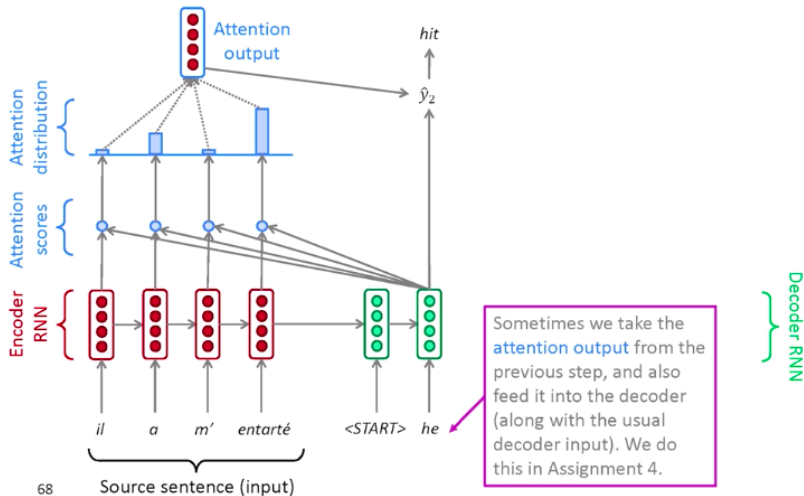
# Attention

## Sequence-to-sequence with attention



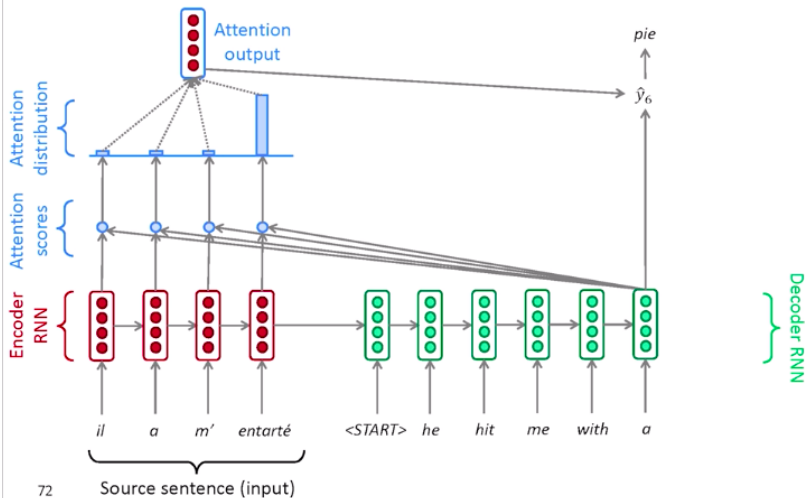
# Attention

## Sequence-to-sequence with attention

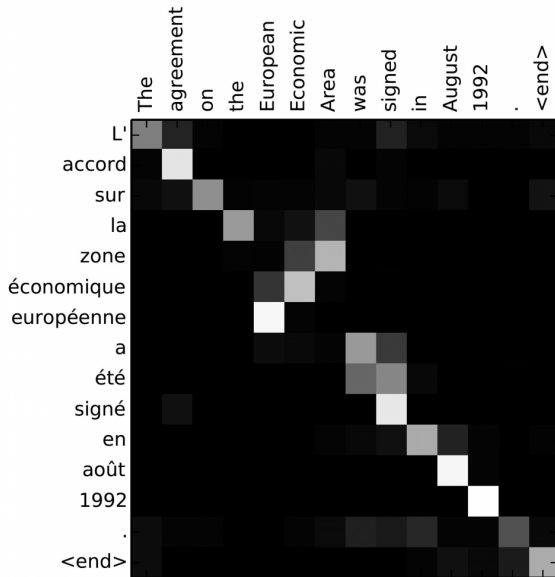


# Attention

## Sequence-to-sequence with attention

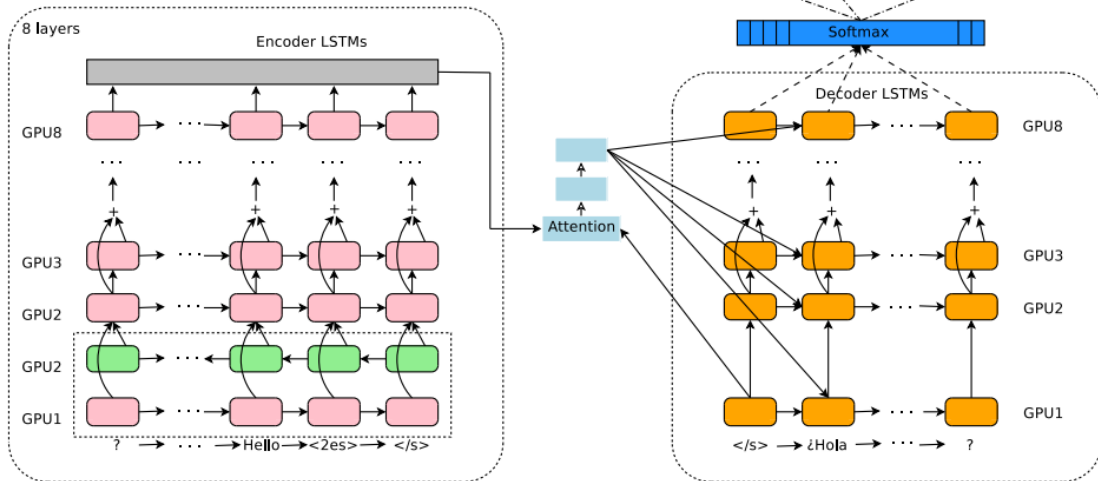


# Attention



# Attention

Идейно - внимание просто выбирает то из эмбедингов, которое действительно нужно для декодирования. Это просто матричное произведение(а можно взвешивать и без весов) и softmax. У нас все остается дифференцируемым - берем градиенты, накапливаем информацию в весах



В целом глобальное решение было найдено, осталось закидать проблему железом.

Выводы:

1. 8 слоев LSTM (8 Карл!)
2. в attention 2 слоя dense.
3. Собираем слова из морфем - пытаемся победить out-of-vocabulary.
4. Модель стала иногда сексистом и фашистом - требуются слишком большие дата сет, чтобы учить эту большую прелесть.

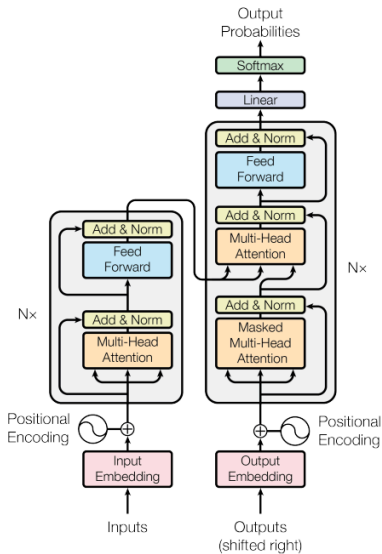
Attention is all you need!



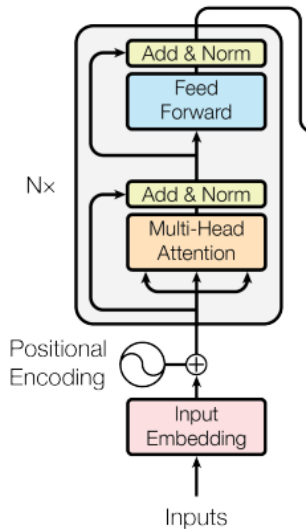
# attention is all you need

- Развитие идеи внимания. Статья вышла в 2017 году и стала прародиелем всех текущих SOTA моделей.
- А зачем нам вообще что-то, кроме внимания?
- Давайте напишем в энкодер и декодер как можно больше внимания и будем такой штукой его учить.

# attention is all you need

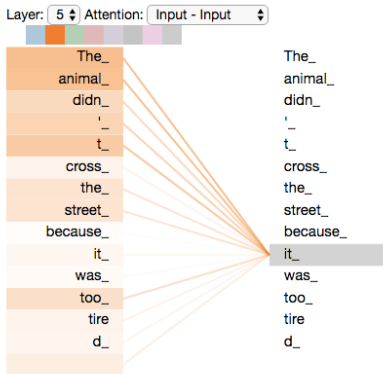


# Encoder



# Что мы хотим?

Есть предложение: "The animal didn't cross the street because it was too tired"



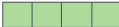
# Абстракции!

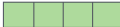
Input

Thinking


Machines


Embedding

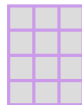
$x_1$  

$x_2$  

Queries

$q_1$  

$q_2$  



$W^Q$

Keys

$k_1$  

$k_2$  



$W^K$

Values

$v_1$  

$v_2$  



$W^V$

А теперь тоже самое, но словами:

1. Query, key - ищем связи между словами. Ходим по всем со всеми смотрим насколько они связаны. Query - мое текущее слово, key - мое слово с которым я сравниваю себя.
2. Value - то, что мы знаем об этом слове

# Mar 2

Input

Embedding

Queries

Keys

Values

Score

Thinking

$x_1$  

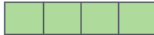
$q_1$  

$k_1$  

$v_1$  

$$q_1 \cdot k_1 = 112$$

Machines

$x_2$  

$q_2$  

$k_2$  

$v_2$  

$$q_1 \cdot k_2 = 96$$

# Mar 3

Input

Embedding

Queries

Keys

Values

Score

Divide by 8 (  $\sqrt{d_k}$  )

Softmax

Thinking

$x_1$  

$q_1$  

$k_1$  

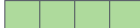
$v_1$  

$q_1 \cdot k_1 = 112$

14

0.88

Machines

$x_2$  

$q_2$  

$k_2$  

$v_2$  

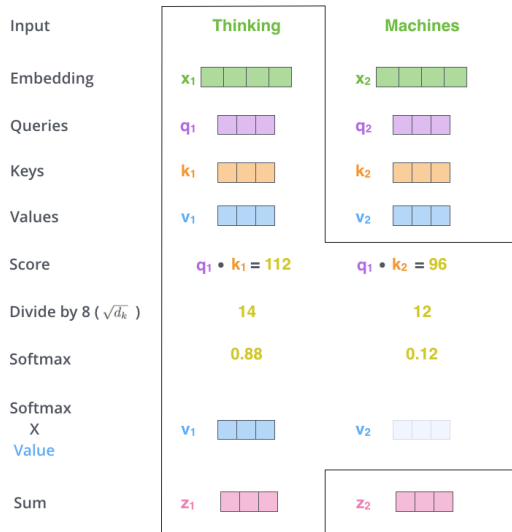
$q_2 \cdot k_2 = 96$

12

0.12



# Mar 4



# Шаг 5

$$\mathbf{X} \times \mathbf{W}^Q = \mathbf{Q}$$


$$\mathbf{X} \times \mathbf{W}^K = \mathbf{K}$$


$$\mathbf{X} \times \mathbf{W}^V = \mathbf{V}$$


# Подробнее

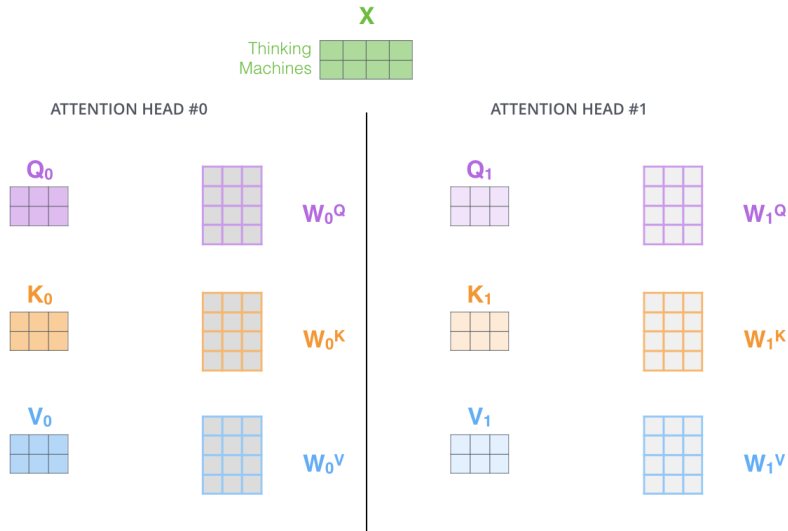
$$\text{softmax} \left( \frac{\begin{matrix} \textcolor{violet}{Q} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} \textcolor{brown}{K}^T \\ \begin{array}{|c|c|} \hline & \\ \hline & \\ \hline & \\ \hline \end{array} \end{matrix} \right) \begin{matrix} \textcolor{blue}{V} \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

$=$

$\textcolor{pink}{Z}$

$\begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline \end{array}$

# multi head attention



# Соединяем!

1) Concatenate all the attention heads

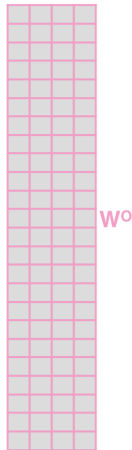


3) The result would be the  $Z$  matrix that captures information from all the attention heads. We can send this forward to the FFNN



2) Multiply with a weight matrix  $W^O$  that was trained jointly with the model

$\times$



# Проблема

При таком подходе теряется расстояние между словами

# Position Encoding

$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k + 1 \end{cases}$$

where

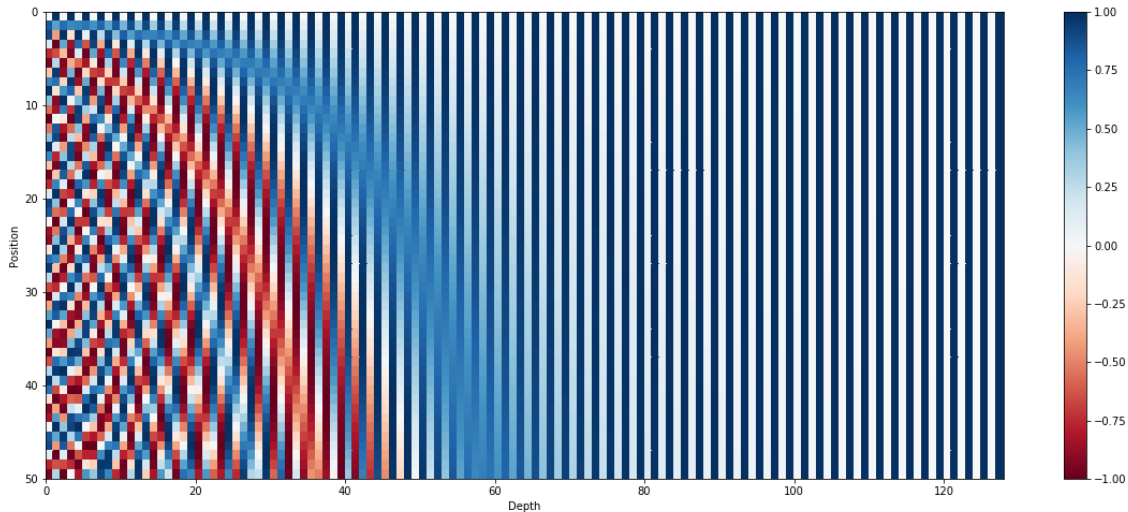
$$\omega_k = \frac{1}{10000^{2k/d}}$$

# Position Encoding

$$\vec{p_t} = \begin{bmatrix} \sin(\omega_1 \cdot t) \\ \cos(\omega_1 \cdot t) \\ \\ \sin(\omega_2 \cdot t) \\ \cos(\omega_2 \cdot t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} \cdot t) \\ \cos(\omega_{d/2} \cdot t) \end{bmatrix}_{d \times 1}$$



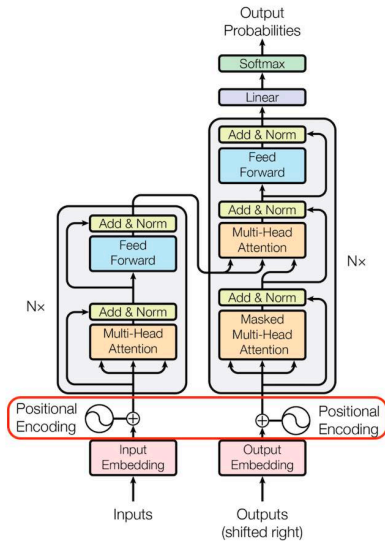
# Position Encoding



# Position Encoding

$$M. \begin{bmatrix} \sin(\omega_k \cdot t) \\ \cos(\omega_k \cdot t) \end{bmatrix} = \begin{bmatrix} \sin(\omega_k \cdot (t + \phi)) \\ \cos(\omega_k \cdot (t + \phi)) \end{bmatrix}$$

# Position Encoding



# Итого

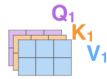
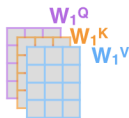
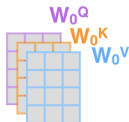
1) This is our input sentence\*

Thinking  
Machines

2) We embed each word\*



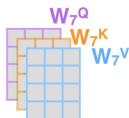
3) Split into 8 heads.  
We multiply  $X$  or  $R$  with weight matrices



...

...

...



4) Calculate attention using the resulting  $Q/K/V$  matrices

5) Concatenate the resulting  $Z$  matrices, then multiply with weight matrix  $W^O$  to produce the output of the layer

$W^O$



$Z$

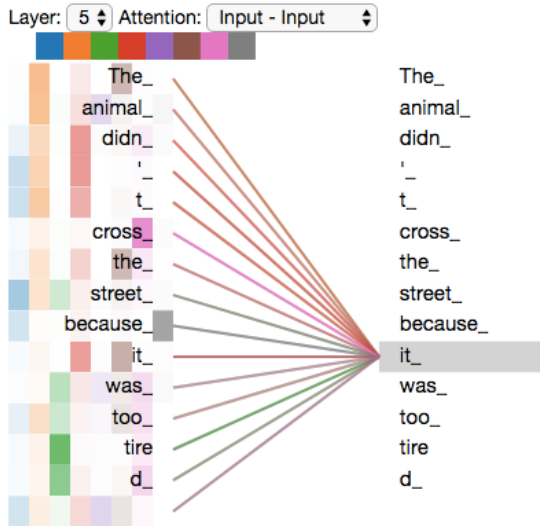


\* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

$R$



# Итого



# Итого

1. У нас нет никаких слоев, кроме линейных
2. Работает в 3 раза быстрее LSTM
3. Можно кормить последовательности произвольной длины
4. Учится очень классно, находит множество взаимосвязей
5. позицион энкодинг позволяет учитывать позицию в тексте

# ELMO (Embeddings from Language Model)

# Семантика слова ЗАВИСИТ ОТ КОНТЕКСТА

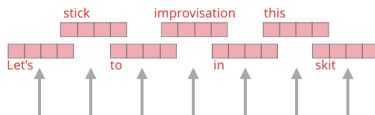




# ELMo

Захватываем контекст предложения через biderictional LSTM. Таким образом мы захватываем и контекст предложения

ELMo  
Embeddings

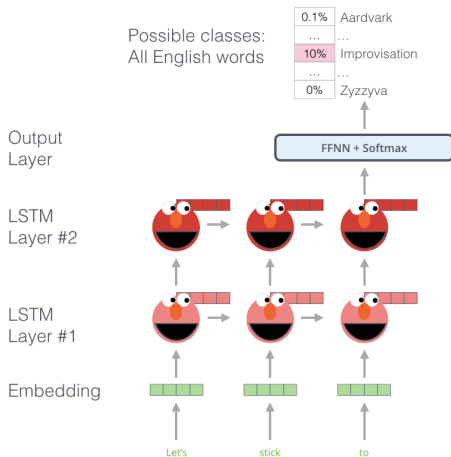


Words to embed



# ELMO

Учится понимать язык ELMO следующим образом - оно берет большой датасет и пытается предсказать следующее слово в предложении.



## Применяем

## Embedding of "stick" in "Let's stick to" - Step #2

1- Concatenate hidden layers



2- Multiply each vector by a weight based on the task

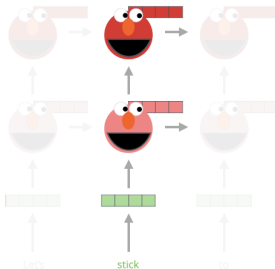


3- Sum the (now weighted) vectors

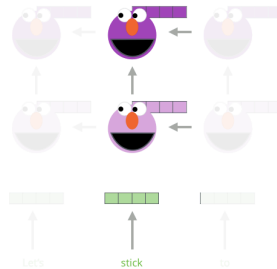


ELMo embedding of "stick" for this task in this context

Forward Language Model



Backward Language Model



## Применяем

## Embedding of "stick" in "Let's stick to" - Step #2

1- Concatenate hidden layers



2- Multiply each vector by a weight based on the task

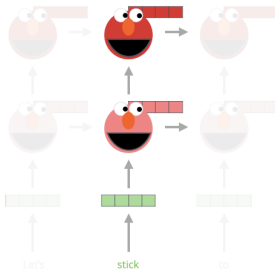


3- Sum the (now weighted) vectors

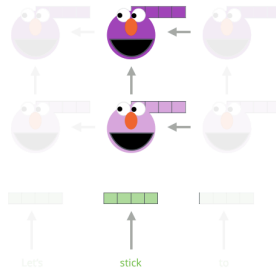


ELMo embedding of "stick" for this task in this context

Forward Language Model



Backward Language Model



# 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.

## Semi-supervised Learning Step

**Model:**



**Dataset:**



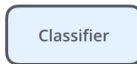
**Objective:**

Predict the masked word  
(language modeling)

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

## Supervised Learning Step

**Model:**  
(pre-trained  
in step #1)



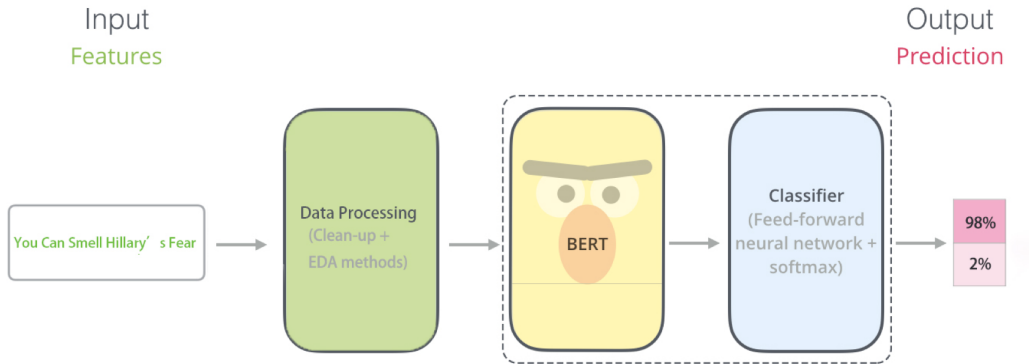
75% Spam  
25% Not Spam



**Dataset:**

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached...	Not Spam

# BERT



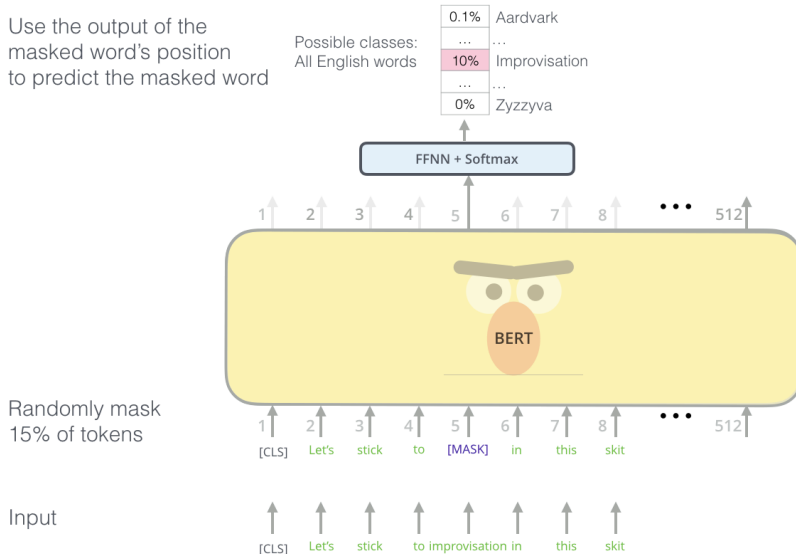
# Лежащие внутри идеи и почему он популярный

1. Предобучаем по двум задачам - берем корпус текстов и маскируем часть предложений, заставляем учить и предсказывать маску.
2. И вторая идея - предсказываем следующее слово в предложении
3. Он из коробки знает язык, ему 1-2 эпохи надо подсказать, что с этим знанием делать
4. В готовых либах лежат много готовых под задачи бертов - классификация, вопросно - ответные системы и тому подобное.
5. Опять же - подаем слова кусочками, чтобы как-то решать проблему оов.



# Предобучение

Use the output of the masked word's position to predict the masked word



# Как используем?

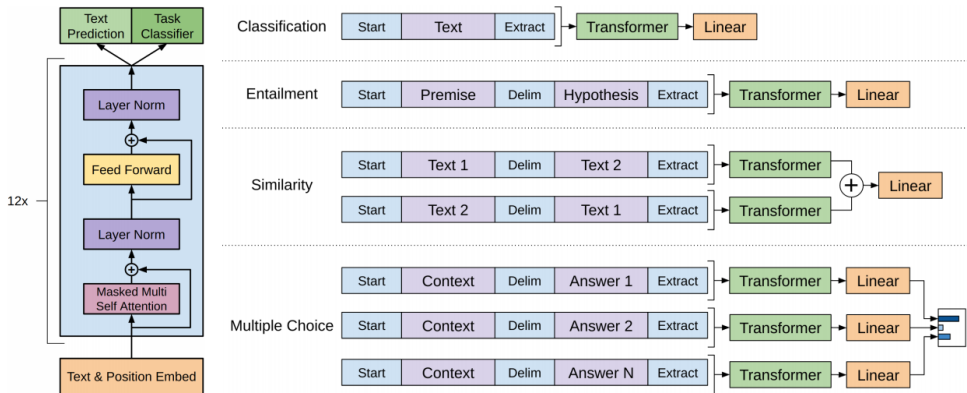


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

# Растём в ширь

BERT - количество параметров: 345 миллионов

GPT-3 количество параметров: ...

# Растём в ширь

BERT - количество параметров: 345 миллионов

GPT-3 количество параметров: **175 миллиарда**

# Linformer

Трансформер  $O(n^2)$  от длины последовательности

# Linformer

Хитрыми математическими преобразованиями, можно добиться линейной сложности  $O(n)$  при этом не сильно потеряв в качестве

# Linformer

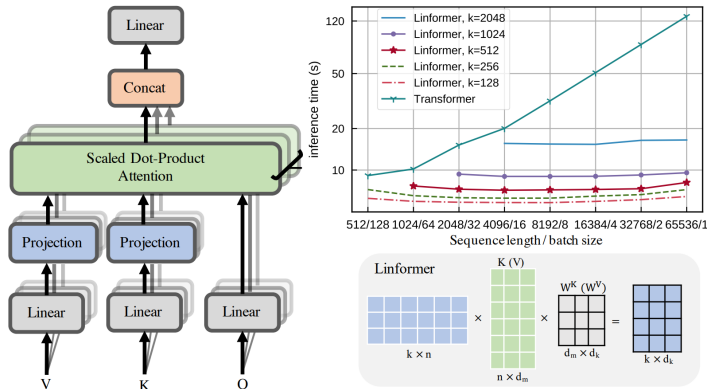


Figure 2: Left and bottom-right show architecture and example of our proposed multihead linear self-attention. Top right shows inference time vs. sequence length for various Linformer models.

Статья: <https://arxiv.org/pdf/2006.04768.pdf>