

# Машинное обучение

## *Lecture 1 - Введение в NLP*

*Власов Кирилл Вячеславович*



2021

**Обработка естественного языка** (*Natural Language Processing, NLP*) – общее направление искусственного интеллекта и математической лингвистики.

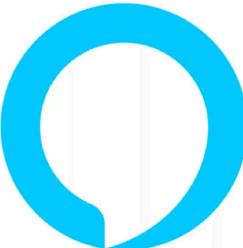
**Обработка естественного языка** (*Natural Language Processing, NLP*) – общее направление искусственного интеллекта и математической лингвистики.



**Обработка естественного языка** (*Natural Language Processing, NLP*) – общее направление искусственного интеллекта и математической лингвистики.



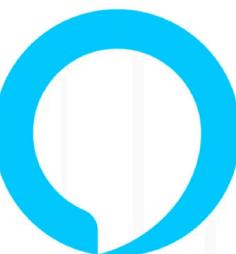
**Обработка естественного языка** (*Natural Language Processing, NLP*) – общее направление искусственного интеллекта и математической лингвистики.



**Обработка естественного языка (Natural Language Processing, NLP)** – общее направление искусственного интеллекта и математической лингвистики.

Яндекс

Google



- Распознавание речи (ASR)
- Анализ текста
  - Извлечение информации
  - Информационный поиск
  - Анализ высказываний
  - Анализ тональности текста
  - Вопросно-ответные системы
- Генерация текста
- Синтез речи (TTS)
- Задачи анализа и синтеза в комплексе
- Машинный перевод
- Автоматическое реферирование, аннотирование или упрощение текста

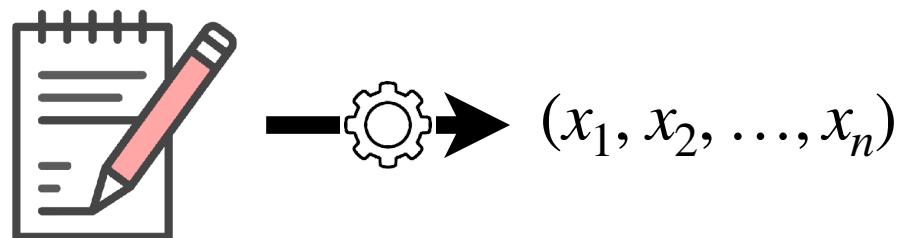
# Еще примеры и задачи

- Дискретные метки (классификация)
  - Бинарные
    - *Спам-фильтр*
    - *Анализ тональности*
  - Multi-class
    - *классификация по категориям*
  - Multi-label
    - *Предсказание хэштегов*
- Вещественные метки (регрессия):
  - *Предсказание цены по описанию товара*
  - *Предсказание зарплаты по резюме*

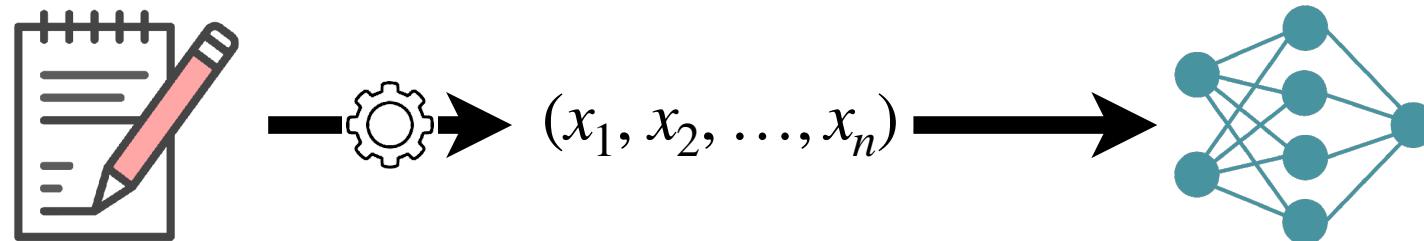
# Как решить задачу в общем виде?



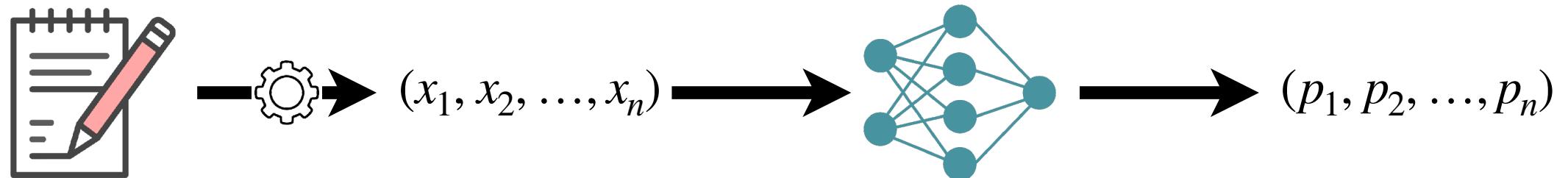
# Как решить задачу в общем виде?



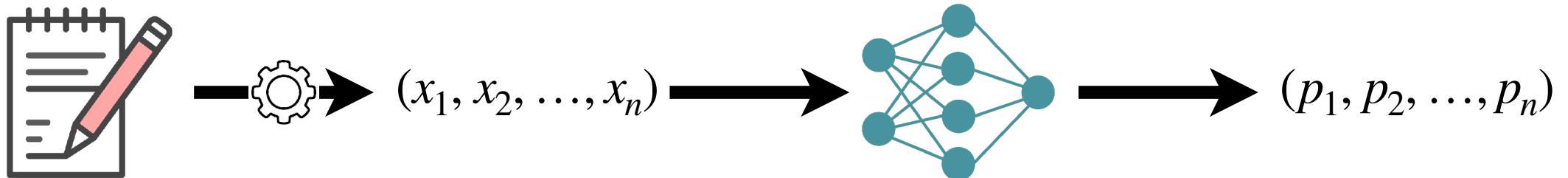
# Как решить задачу в общем виде?



# Как решить задачу в общем виде?

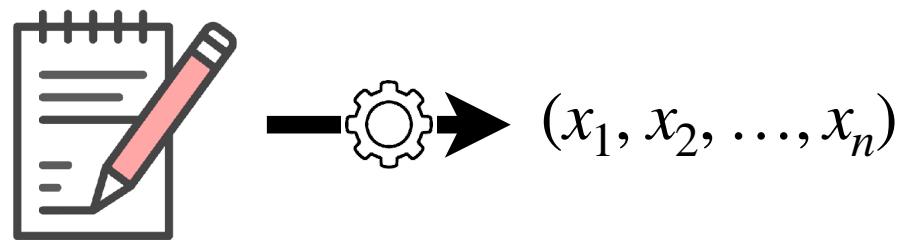


# Как решить задачу в общем виде?



Ну, ПРИВЕТ





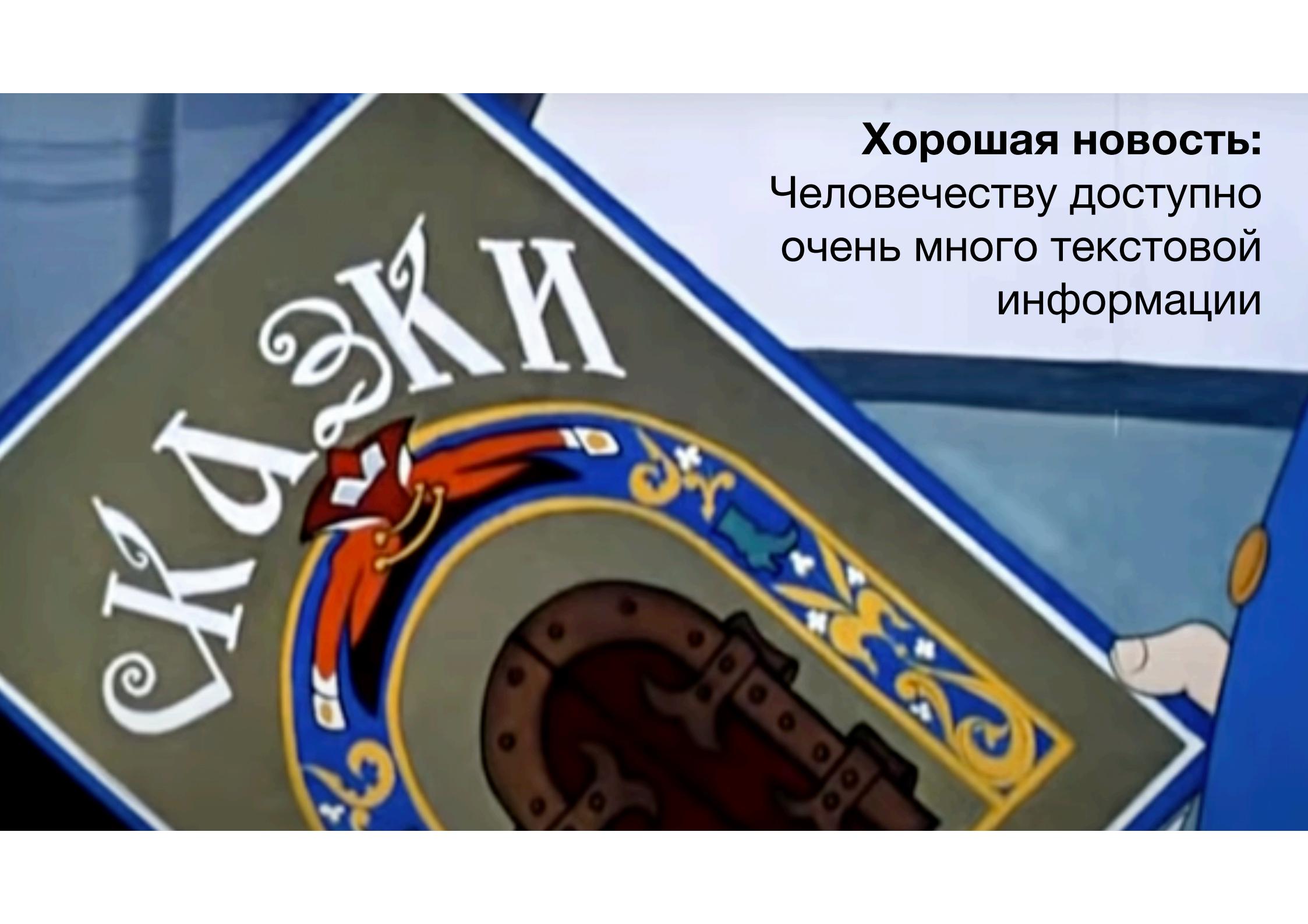
Почему задача нетривиальна?

# Почему задача нетривиальна?

Перед нами стол. На столе стакан и вилка. Что они делают?

Стакан стоит, а вилка лежит. Если мы воткнем вилку в столешницу, вилка будет стоять. Т.е. стоят вертикальные предметы, а лежат горизонтальные? Добавляем на стол тарелку и сковороду. Они вроде как горизонтальные, но на столе стоят. Теперь положим тарелку в сковородку. Там она лежит, а ведь на столе стояла. Может быть, стоят предметы готовые к использованию? Нет, вилка-то готова была, когда лежала. Теперь на стол залезает кошка. Она может стоять, сидеть и лежать. Если в плане стояния и лежания она как-то лезет в логику «вертикальный-горизонтальный» , то сидение - это новое свойство. Сидит она на попе. Теперь на стол села птичка. Она на столе сидит, но сидит на ногах, а не на попе. Хотя вроде бы должна стоять. Но стоять она не может вовсе. Но если мы убьём бедную птичку и сделаем чучело, оно будет на столе стоять. Может показаться, что сидение - атрибут живого, но сапог на ноге тоже сидит, хотя он не живой и не имеет попы. Так что, поди ж пойми, что стоит, что лежит, а что сидит.





**Хорошая новость:**  
Человечеству доступно  
очень много текстовой  
информации

# Препроцессинг

# Препроцессинг

**Исходный документ из  
корпуса текстов**



На практике очень часто возникают задачи для решения которых используются методы оптимизации в обычной жизни при множественном выборе например подарков к новому году мы интуитивно решаем задачу минимальных затрат при заданном качестве покупок

# Препроцессинг

**Исходный документ из  
корпуса текстов**

На практике очень часто возникают задачи для решения которых используются методы оптимизации в обычной жизни при множественном выборе например подарков к новому году мы интуитивно решаем задачу минимальных затрат при заданном качестве покупок

на практике очень часто возникать задача для решения который использоваться метод оптимизация в обычный жизнь при множественный выбор например подарок к новый год мы интуитивно решать задача минимальный затраты при задавать качество покупка

**Лемматизация**

# Препроцессинг

## Исходный документ из корпуса текстов

На практике очень часто возникают задачи для решения которых используются методы оптимизации в обычной жизни при множественном выборе например подарков к новому году мы интуитивно решаем задачу минимальных затрат при заданном качестве покупок

на практике очень часто возникает задача для решения который используется метод оптимизация в обычный жизнь при множественный выбор например подарок к новый год мы интуитивно решать задача минимальный затраты при задавать качество покупка

## Лемматизация

## Стеминг

на практик очень част возника задачи, для решен котор использ метод оптимизац в обычн жизн при множествен выборе, например, подарк к нов год мы интуитивн решает задача минимальн затрат при зада качеств покупок

# Препроцессинг

## Исходный документ из корпуса текстов

На практике очень часто возникают задачи для решения которых используются методы оптимизации в обычной жизни при множественном выборе например подарков к новому году мы интуитивно решаем задачу минимальных затрат при заданном качестве покупок

## Стеминг

на практик очень част возника задачи, для решен котор использ метод оптимизац в обычн жизн при множествен выборе, например, подарк к нов год мы интуитивн реша задач минимальн затрат при зада качеств покупок

на практика очень часто возникать задача для решение который использоваться метод оптимизация в обычный жизнь при множественный выбор например подарок к новый год мы интуитивно решать задача минимальный затрата при задавать качество покупка

## Лемматизация



# Feature engineering

Использовать ли короткие слова или стоит их выкинуть? (предлоги)

Использовать ли очень частые и/или очень редкие?

Какой длины использовать Ngrams? Нужны ли они?

**Шум в данных**

**Огромный объем  
данных**

## Recap: Ngrams

практика очень часто возникать задача

2grams

практика очень  
очень часто  
часто возникать  
возникать задача

# Feature engineering

## Bag of words

Три документа:

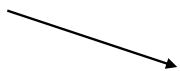
- Артур любит смотреть фильмы
- Ольга тоже любит фильмы
- Денис любит смотреть футбол

# Feature engineering

## Bag of words

Три документа:

- Артур любит смотреть фильмы
- Ольга тоже любит фильмы
- Денис любит смотреть футбол



Артур любит смотреть фильмы  
Ольга тоже Денис футбол



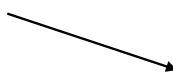
# Feature engineering

## Bag of words

Три документа:

- Артур любит смотреть фильмы
- Ольга тоже любит фильмы
- Денис любит смотреть футбол

Артур любит смотреть фильмы  
Ольга тоже Денис футбол



id	Артур	Ольга	Денис	Любит	смотреть	Фильмы	тоже	Футбол
1	1	0	0	1	1	1	0	0
2	0	1	0	1	0	1	1	0
3	0	0	1	1	1	0	0	1

# Feature engineering

**TF-IDF** (от англ. *TF* — *term frequency*, *IDF* — *inverse document frequency*) — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.

# Feature engineering

**TF** (term frequency — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

$$\text{tf}(t, d) = \frac{n_t}{\sum_k n_k} ,$$

где  $n_t$  есть число вхождений слова  $t$  в документ, а в знаменателе — общее число слов в данном документе.

# Feature engineering

**TF** (term frequency — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

$$\text{tf}(t, d) = \frac{n_t}{\sum_k n_k} ,$$

где  $n_t$  есть число вхождений слова  $t$  в документ, а в знаменателе — общее число слов в данном документе.

**IDF** (inverse document frequency — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции.

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|} , [2]$$

где

- $|D|$  — число документов в коллекции;
- $|\{d_i \in D \mid t \in d_i\}|$  — число документов из коллекции  $D$ , в которых встречается  $t$  (когда  $n_t \neq 0$ ).

# Feature engineering

**TF** (term frequency — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

$$\text{tf}(t, d) = \frac{n_t}{\sum_k n_k} ,$$

где  $n_t$  есть число вхождений слова  $t$  в документ, а в знаменателе — общее число слов в данном документе.

**IDF** (inverse document frequency — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции.

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|} , [2]$$

где

- $|D|$  — число документов в коллекции;
- $|\{d_i \in D \mid t \in d_i\}|$  — число документов из коллекции  $D$ , в которых встречается  $t$  (когда  $n_t \neq 0$ ).

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

**Большой вес в TF-IDF получат слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.**





NLTK

```
from nltk.stem.snowball import SnowballStemmer  
from nltk.stem.porter import PorterStemmer  
from nltk.stem import WordNetLemmatizer  
from nltk.corpus import stopwords
```





## NLTK

```
from nltk.stem.snowball import SnowballStemmer  
from nltk.stem.porter import PorterStemmer  
from nltk.stem import WordNetLemmatizer  
from nltk.corpus import stopwords
```

## beautifulsoup





## NLTK

```
from nltk.stem.snowball import SnowballStemmer  
from nltk.stem.porter import PorterStemmer  
from nltk.stem import WordNetLemmatizer  
from nltk.corpus import stopwords
```

## beautifulsoup

## pymorphy2





## NLTK

```
from nltk.stem.snowball import SnowballStemmer  
from nltk.stem.porter import PorterStemmer  
from nltk.stem import WordNetLemmatizer  
from nltk.corpus import stopwords
```

## beautifulsoup

## pymorphy2

## sklearn

```
feature_extraction.text.CountVectorizer(*  
[, ...])  
feature_extraction.text.HashingVectorizer()  
feature_extraction.text.TfidfTransformer()  
feature_extraction.text.TfidfVectorizer(*  
[, ...])
```



## One-hot encoding:

Rome = [1, 0, 0, 0, 0, 0, ..., 0]  
Paris = [0, 1, 0, 0, 0, 0, ..., 0]  
Italy = [0, 0, 1, 0, 0, 0, ..., 0]  
France = [0, 0, 0, 1, 0, 0, ..., 0]

The diagram illustrates the one-hot encoding of four words: Rome, Paris, Italy, and France. It shows four horizontal vectors, each representing a word. Above the first vector, 'Rome' is labeled with an arrow pointing to the first element (1). Above the second vector, 'Paris' is labeled with an arrow pointing to the second element (1). Above the third vector, 'Italy' is labeled with an arrow pointing to the third element (1). Above the fourth vector, 'France' is labeled with an arrow pointing to the fourth element (1). The vectors are enclosed in brackets and separated by commas. Ellipses (...) indicate that there are more elements in the vector than shown.

## One-hot encoding:

Rome      = [1, 0, 0, 0, 0, 0, ..., 0]  
Paris     = [0, 1, 0, 0, 0, 0, ..., 0]  
Italy    = [0, 0, 1, 0, 0, 0, ..., 0]  
France = [0, 0, 0, 1, 0, 0, ..., 0]

Rome      Paris      word V

## Проблемы:

- Огромные матрицы
- Сильно разреженные матрицы
- Не учитывается семантика (например синонимы)
- Не учитывается порядок слов



# **Word Embeddings**

# Word Embeddings

Это чё?



# Word2Vec

## Source Text

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

## Training Samples

(the, quick)  
(the, brown)

(quick, the)  
(quick, brown)  
(quick, fox)

(brown, the)  
(brown, quick)  
(brown, fox)  
(brown, jumps)

(fox, quick)  
(fox, brown)  
(fox, jumps)  
(fox, over)

# Word2Vec

## Source Text

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

The quick brown fox jumps over the lazy dog. →

## Training Samples

(the, quick)  
(the, brown)

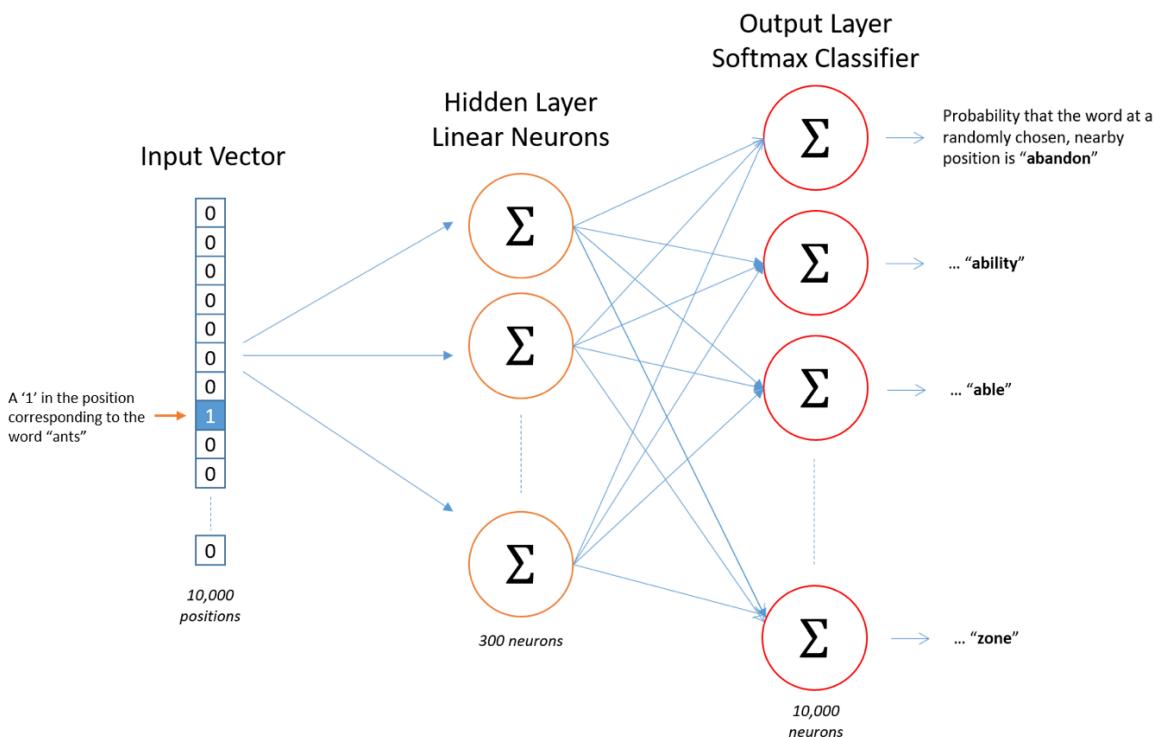
(quick, the)  
(quick, brown)  
(quick, fox)

(brown, the)  
(brown, quick)  
(brown, fox)  
(brown, jumps)

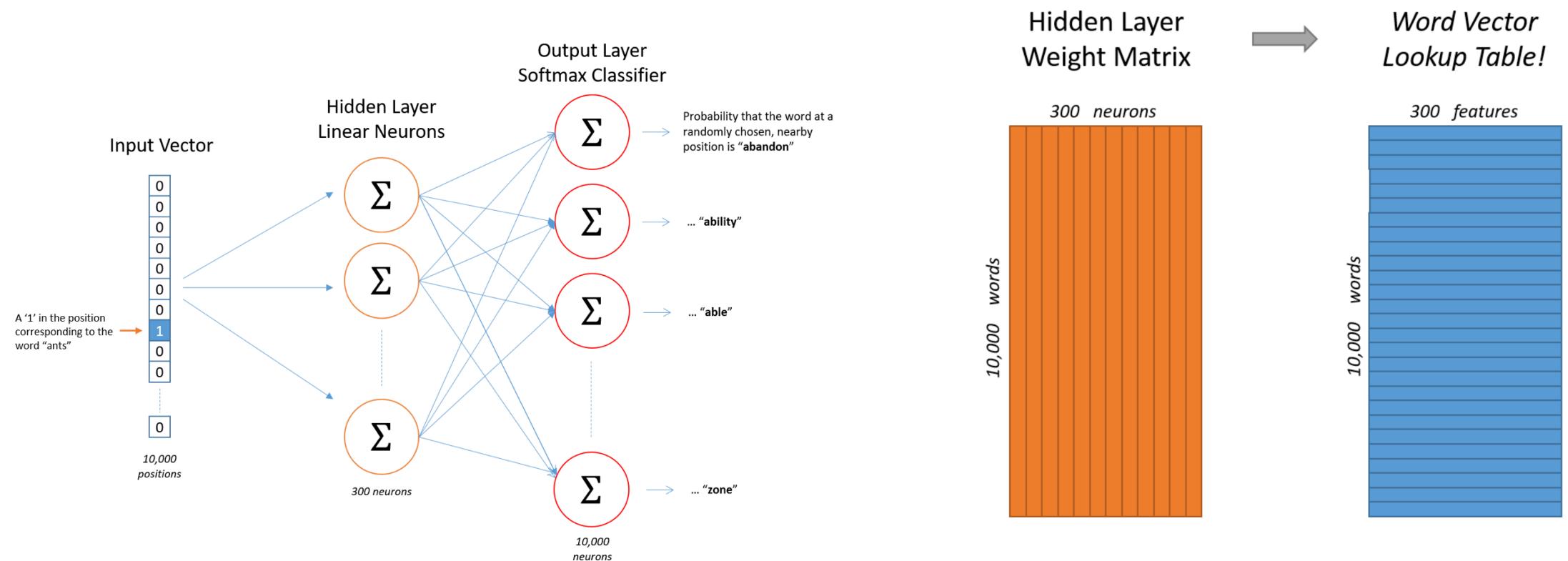
(fox, quick)  
(fox, brown)  
(fox, jumps)  
(fox, over)

**Решим задачу классификации!**

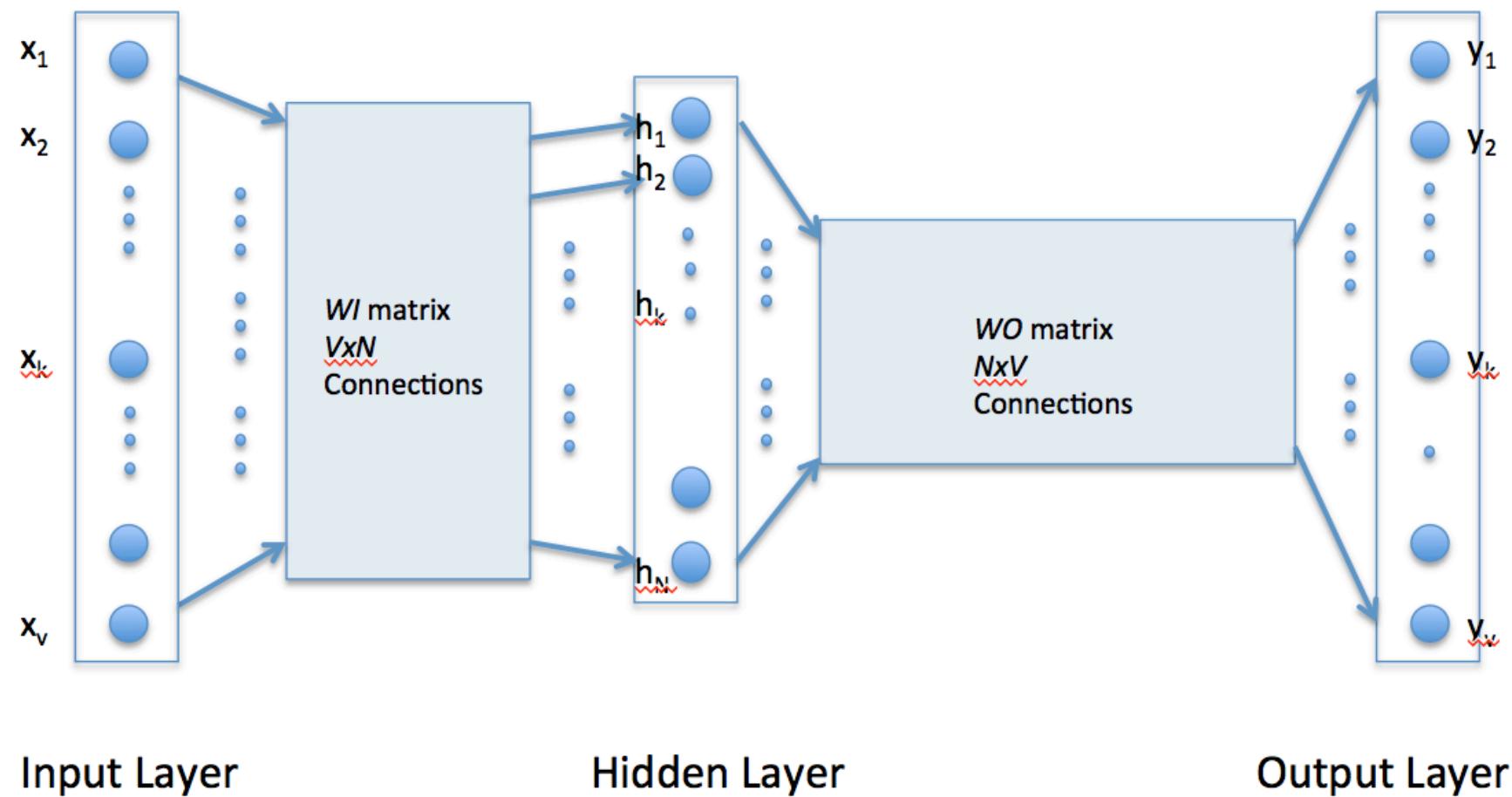
# Word2Vec



# Word2Vec

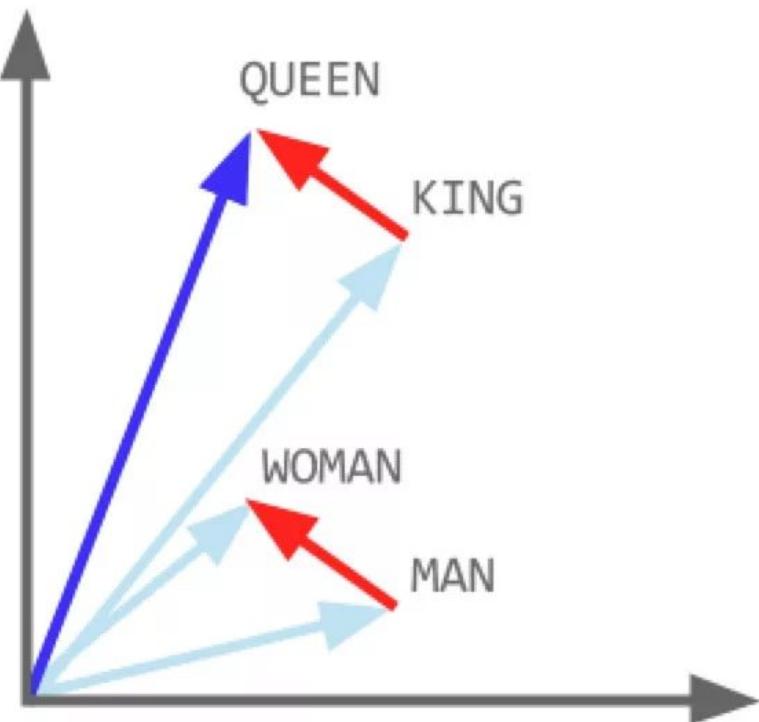


# Word2Vec



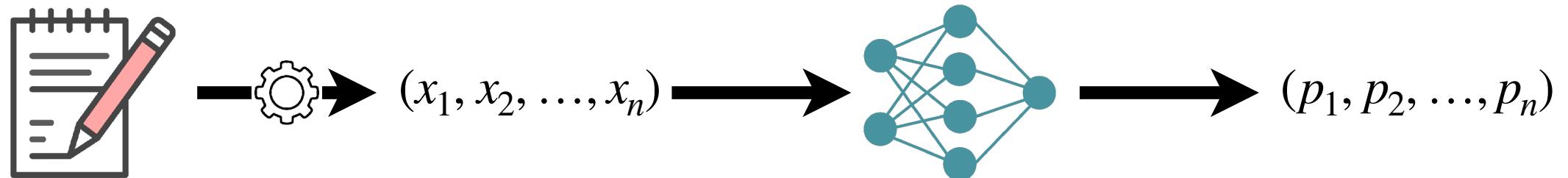
# Word2Vec

So king + man - woman = queen!



- Geopolitics: Iraq - Violence = Jordan
- Distinction: Human - Animal = Ethics
- President - Power = Prime Minister
- Library - Books = Hall
- (Moscow - Russia) + France = ?

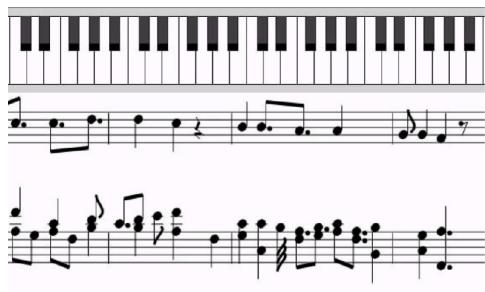
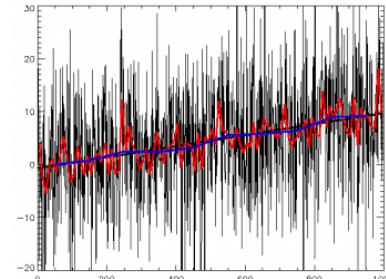
# Как решить задачу в общем виде?



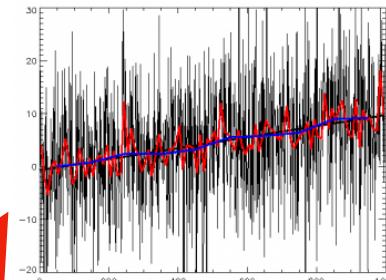
# Последовательности



# Мотивация обработки последовательностей



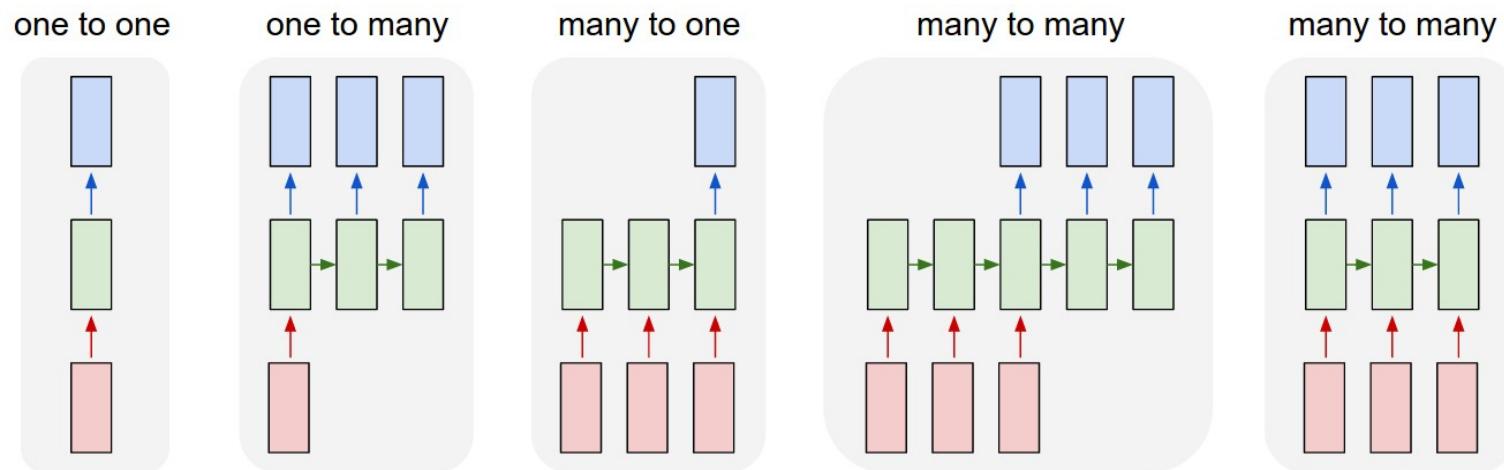
# Мотивация обработки последовательностей



stats: ARIMA/SARIMA/etc.



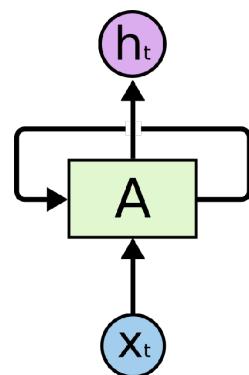
# Мотивация обработки последовательностей



**(1)** Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output (e.g. image classification). **(2)** Sequence output (e.g. image captioning takes an image and outputs a sentence of words). **(3)** Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). **(4)** Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French). **(5)** Synced sequence input and output (e.g. video classification where we wish to label each frame of the video). Notice that in every case are no pre-specified constraints on the lengths sequences because the recurrent transformation (green) is fixed and can be applied as many times as we like.

Andrej Karpathy: [The Unreasonable Effectiveness of Recurrent Neural Networks](#)

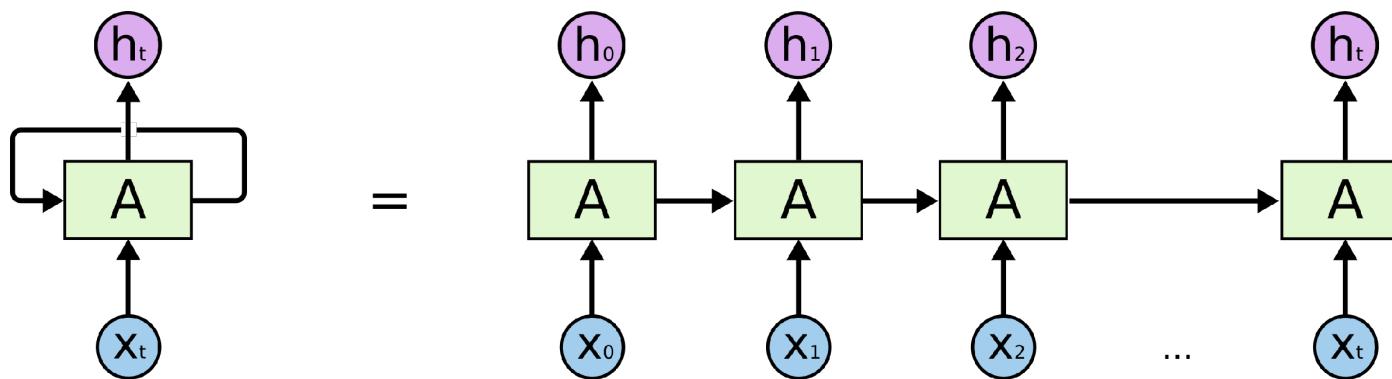
# Рекуррентный нейрон



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t),$$

Christopher Olah: [Understanding LSTM Networks](#)

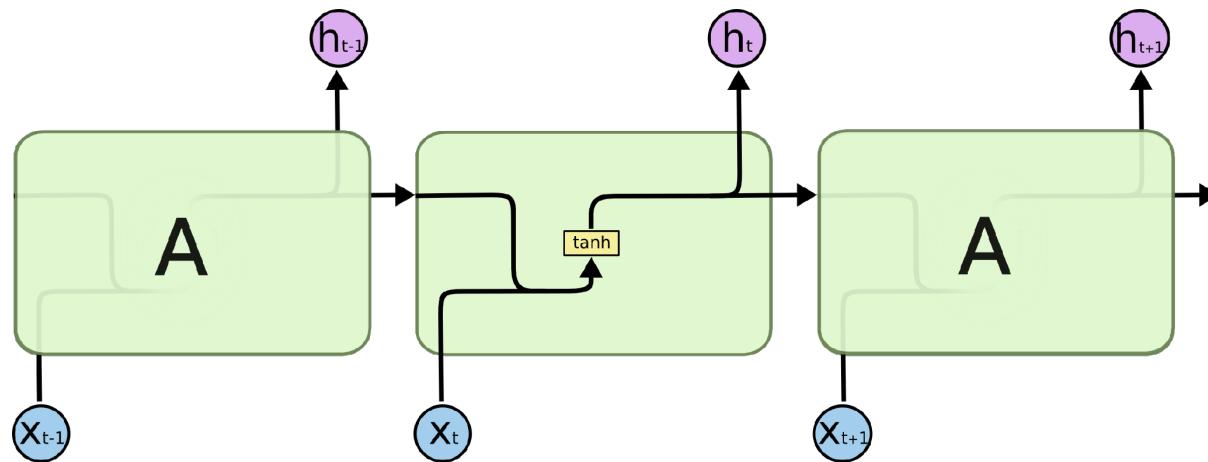
# Рекуррентный нейрон



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t),$$

Christopher Olah: [Understanding LSTM Networks](#)

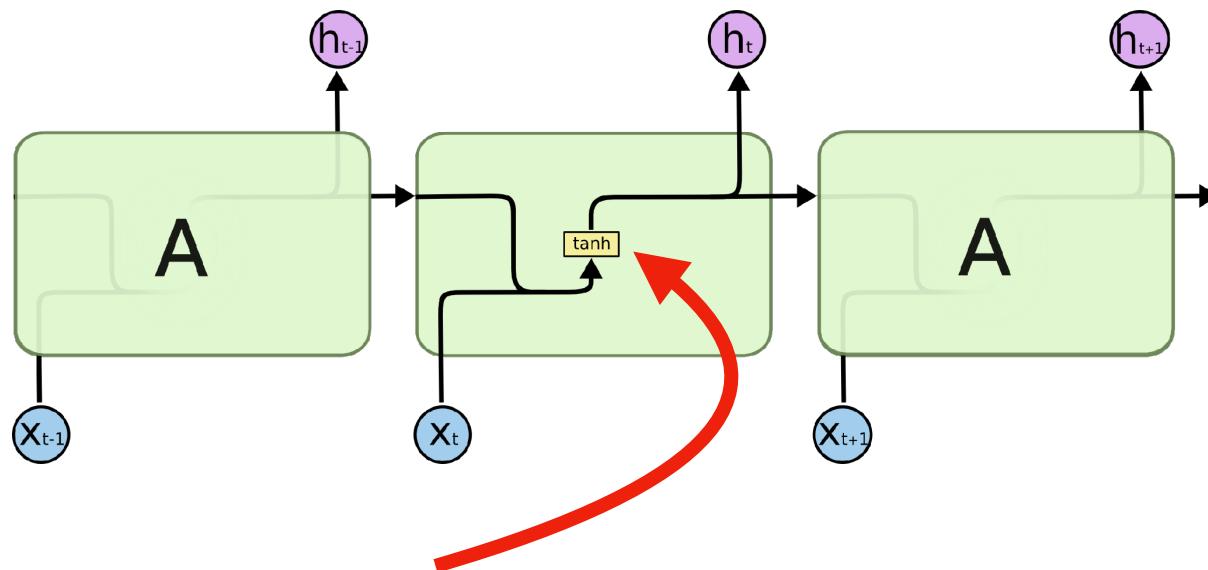
# Vanila RNN



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t),$$

Christopher Olah: [Understanding LSTM Networks](#)

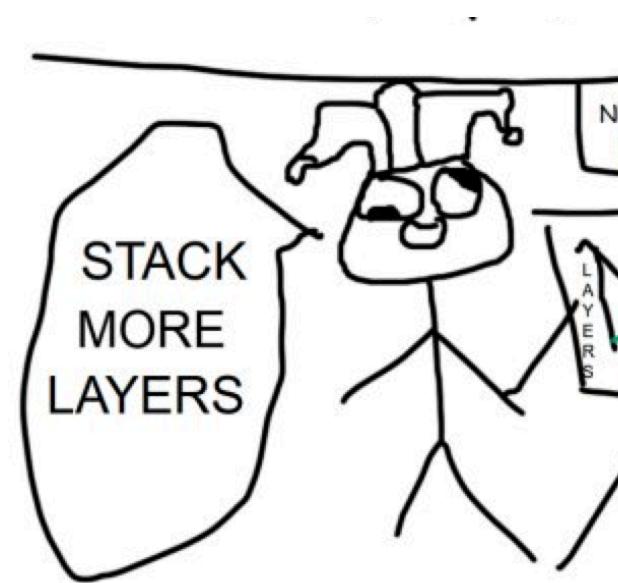
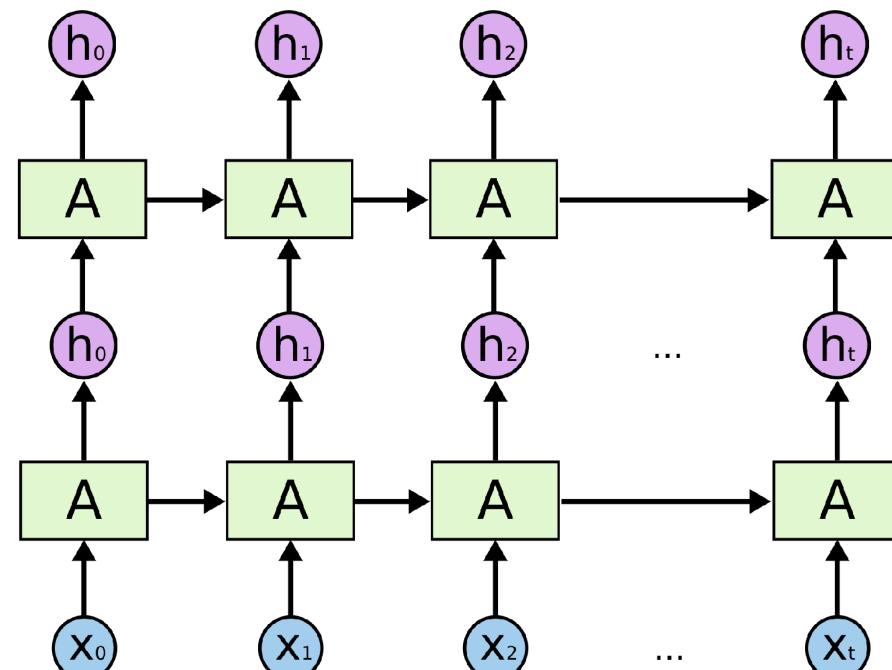
# Vanila RNN



Почему  $\tanh()$ , а не  $\text{sigmoid}()$  или  $\text{ReLU}$ ?

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t),$$

# Больше слоев



# Приимеры

PANDARUS:

Alas, I think he shall be come approached and the day  
When little strain would be attain'd into being never fed,  
And who is but a chain and subjects of his death,  
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,  
Breaking and strongly should be buried, when I perish  
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

They would be ruled after this chamber, and  
my fair nues begun out of the fact, to be conveyed,  
Whose noble souls I'll have the heart of the wars.

Clown:

Come, sir, I will make did behold your worship.

VIOLA:

I'll drink it.

**Lemma 0.1.** Assume (3) and (3) by the construction in the description.

Suppose  $X = \lim |X|$  (by the formal open covering  $X$  and a single map  $\underline{\text{Proj}}_X(\mathcal{A}) = \text{Spec}(B)$  over  $U$  compatible with the complex

$$\text{Set}(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X, \mathcal{O}_X}).$$

When in this case of to show that  $\mathcal{Q} \rightarrow \mathcal{C}_{Z/X}$  is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If  $T$  is surjective we may assume that  $T$  is connected with residue fields of  $S$ . Moreover there exists a closed subspace  $Z \subset X$  of  $X$  where  $U$  in  $X'$  is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem

(1)  $f$  is locally of finite type. Since  $S = \text{Spec}(R)$  and  $Y = \text{Spec}(R)$ .

*Proof.* This is form all sheaves of sheaves on  $X$ . But given a scheme  $U$  and a surjective étale morphism  $U \rightarrow X$ . Let  $U \cap U = \coprod_{i=1, \dots, n} U_i$  be the scheme  $X$  over  $S$  at the schemes  $X_i \rightarrow X$  and  $U = \lim_i X_i$ .  $\square$

The following lemma surjective restrocomposes of this implies that  $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X, \dots, 0}$ .

**Lemma 0.2.** Let  $X$  be a locally Noetherian scheme over  $S$ ,  $E = \mathcal{F}_{X/S}$ . Set  $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$ . Since  $\mathcal{I}^n \subset \mathcal{I}^n$  are nonzero over  $i_0 \leq p$  is a subset of  $\mathcal{J}_{n,0} \circ \bar{A}_2$  works.

**Lemma 0.3.** In Situation ???. Hence we may assume  $q' = 0$ .

*Proof.* We will use the property we see that  $p$  is the next functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where  $K$  is an  $F$ -algebra where  $\delta_{n+1}$  is a scheme over  $S$ .  $\square$

# Приимеры

## SENTIMENT FIXED TO POSITIVE

Just what I was looking for. Nice fitted pants, exactly matched seam to color contrast with other pants I own. Highly recommended and also very happy!

This product does what it is supposed to. I always keep three of these in my kitchen just in case ever I need a replacement cord.

Best hammock ever! Stays in place and holds it's shape. Comfy (I love the deep neon pictures on it), and looks so cute.

Dixie is getting her Doolittle newsletter we'll see another new one coming out next year. Great stuff. And, here's the contents - information that we hardly know about or forget.

I love this weapons look . Like I said beautiful !!! I recommend it to all. Would suggest this to many roleplayers, And I stronge to get them for every one I know. A must watch for any man who love Chess!

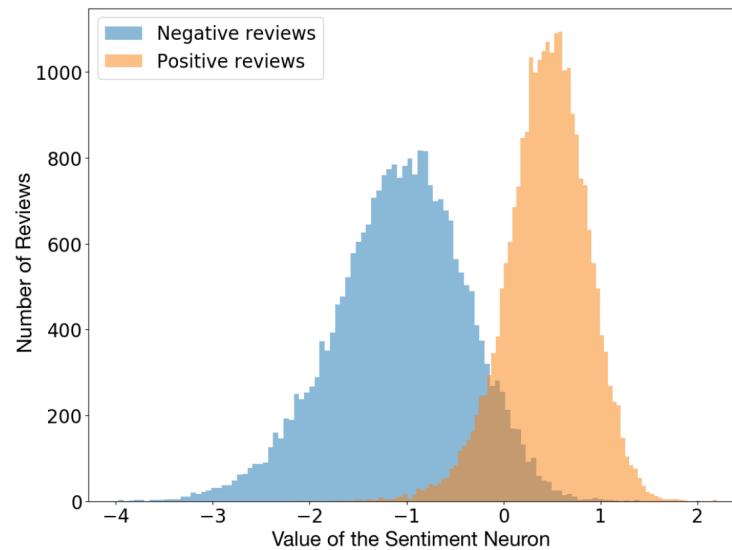
## SENTIMENT FIXED TO NEGATIVE

The package received was blank and has no barcode. A waste of time and money.

Great little item. Hard to put on the crib without some kind of embellishment. My guess is just like the screw kind of attachment I had.

They didn't fit either. Straight high sticks at the end. On par with other buds I have. Lesson learned to avoid.

great product but no seller. couldn't ascertain a cause. Broken product. I am a prolific consumer of this company all the time.



OpenAI: [Unsupervised Sentiment Neuron](#)

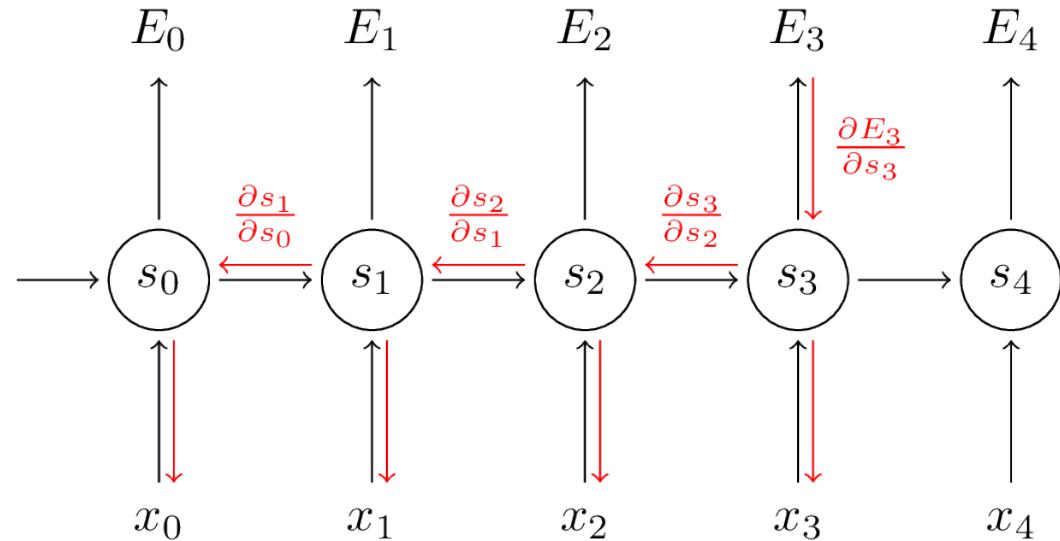
# Обучение RNN

$$s_t = \tanh(Ux_t + Ws_{t-1})$$

$$\hat{y}_t = \text{softmax}(Vs_t)$$

$$E_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t$$

$$\begin{aligned} E(y, \hat{y}) &= \sum_t E_t(y_t, \hat{y}_t) \\ &= - \sum_t y_t \log \hat{y}_t \end{aligned}$$



$$\begin{aligned} \frac{\partial E_3}{\partial V} &= \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial V} \\ &= \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial z_3} \frac{\partial z_3}{\partial V} \\ &= (\hat{y}_3 - y_3) \otimes s_3 \end{aligned}$$

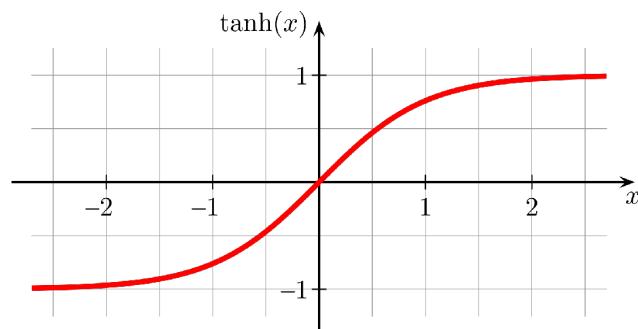
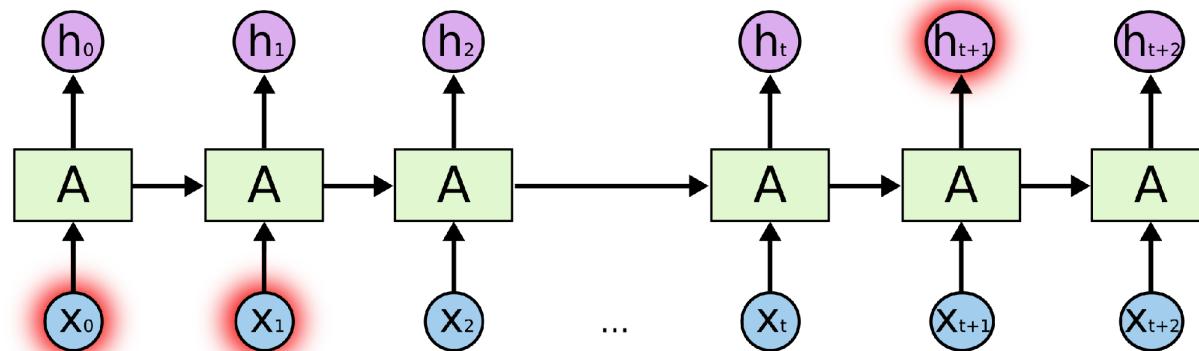
$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial W}$$

**Но:**  $s_3 \leftarrow s_2 \leftarrow s_1$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \frac{\partial s_3}{\partial s_k} \frac{\partial s_k}{\partial W} \implies \frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left( \prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$

Denny Britz: [Backpropagation Through Time and Vanishing Gradients](#)

# Проблема долгих зависимостей

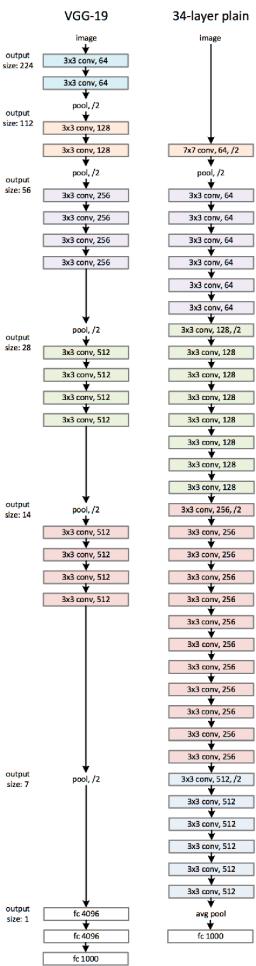


$$s_t = \tanh(Ux_t + Ws_{t-1})$$

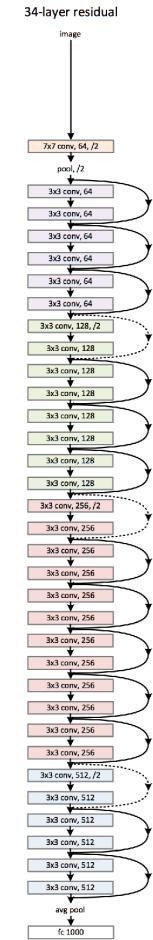
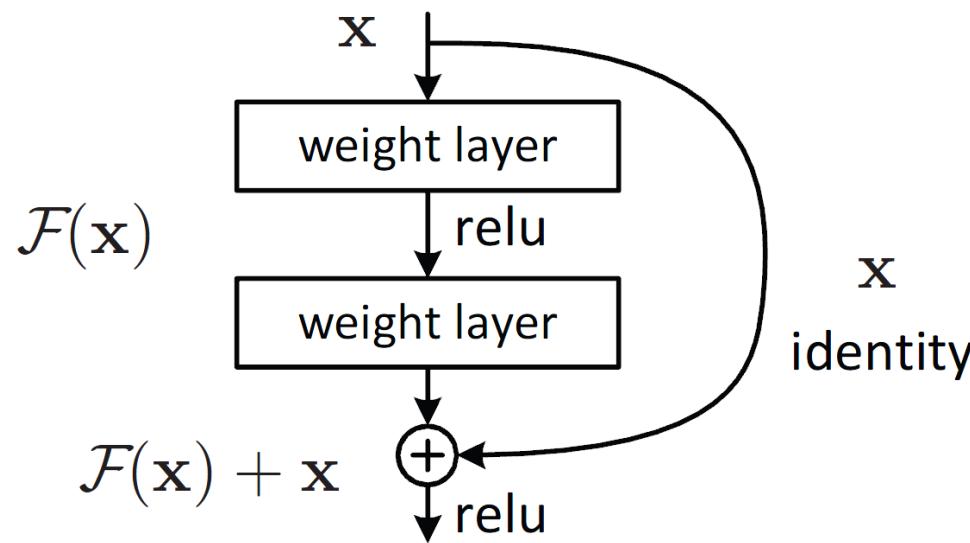
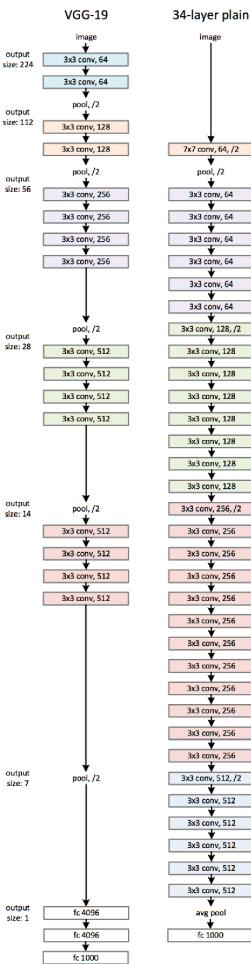
$$\hat{y}_t = \text{softmax}(Vs_t)$$

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^3 \frac{\partial E_3}{\partial \hat{y}_3} \frac{\partial \hat{y}_3}{\partial s_3} \left( \prod_{j=k+1}^3 \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}$$

# Проблема долгих зависимостей

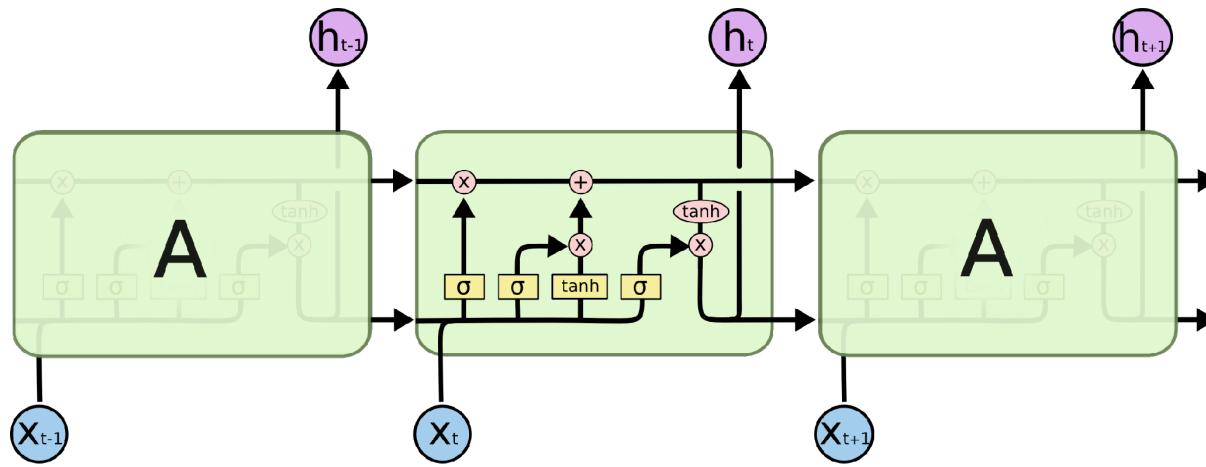


# Проблема долгих зависимостей





# Long Short Term Memory networks

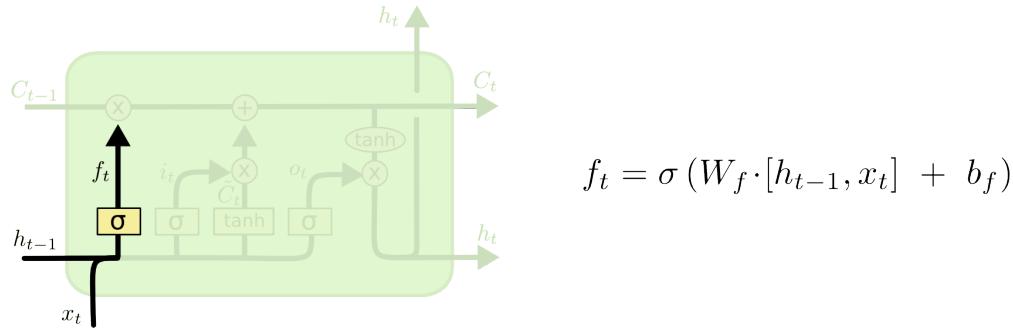


Hochreiter, Sepp & Schmidhuber: [Long Short-term Memory](#)

Christopher Olah: [Understanding LSTM Networks](#)

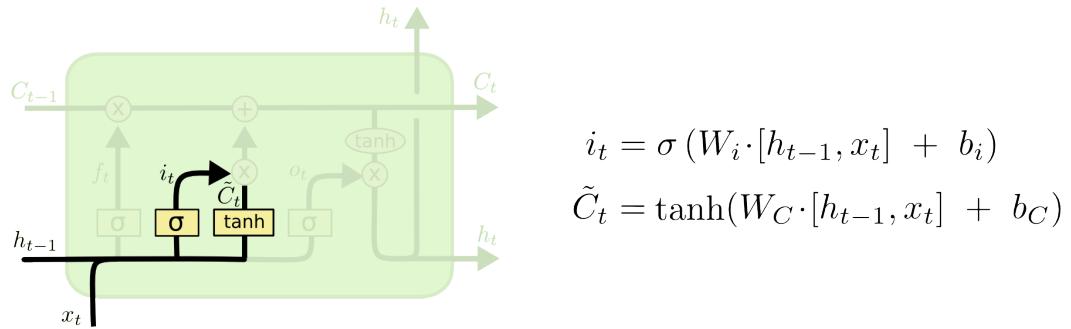
# Long Short Term Memory networks

**Forget gate**



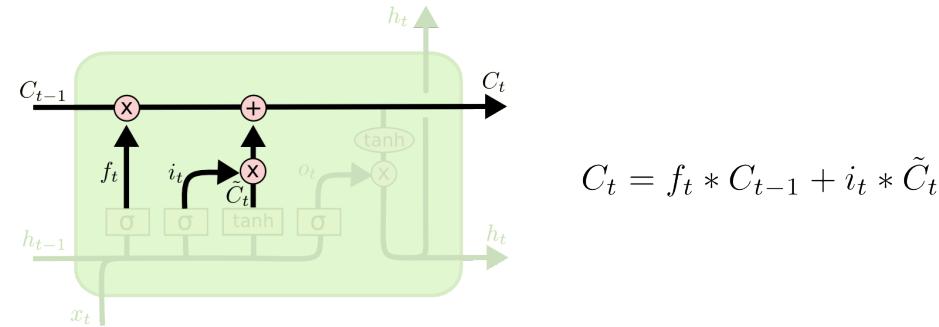
# Long Short Term Memory networks

***Input gate***



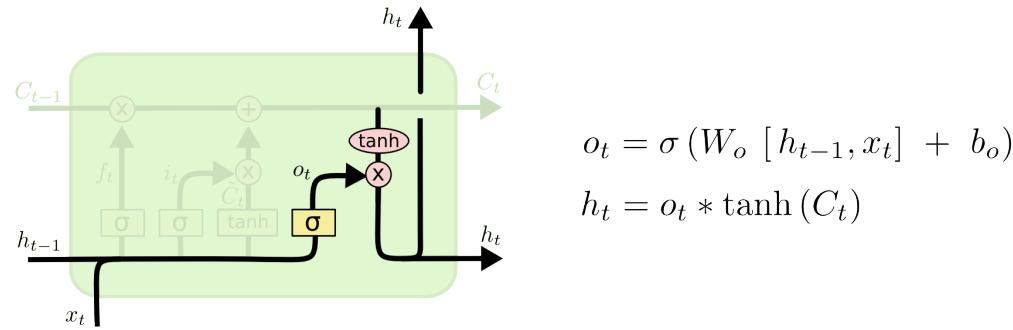
# Long Short Term Memory networks

**Cell update**



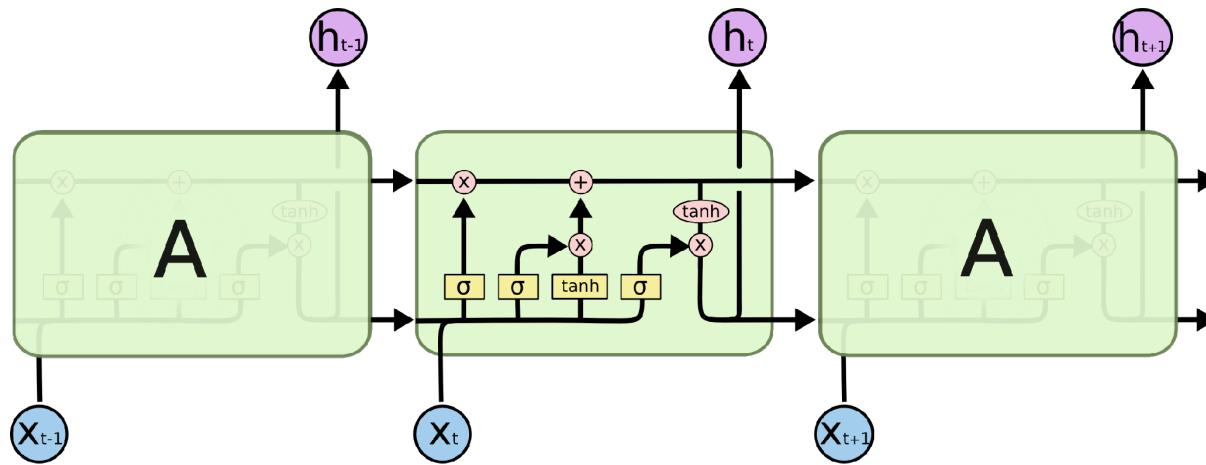
# Long Short Term Memory networks

**Output gate**



Christopher Olah: [Understanding LSTM Networks](#)

# Long Short Term Memory networks

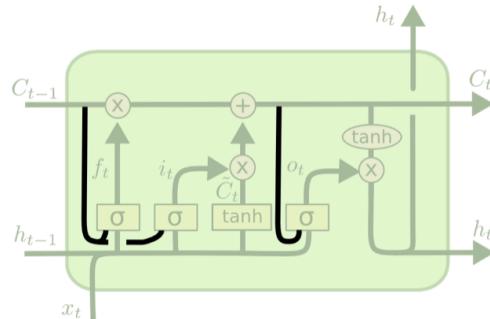


Hochreiter, Sepp & Schmidhuber: [Long Short-term Memory](#)

Christopher Olah: [Understanding LSTM Networks](#)

# Варианты LSTM

**Peephole connections**

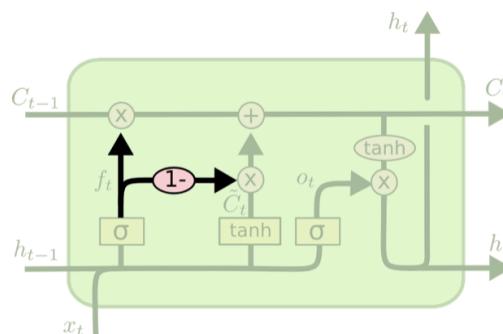


$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

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

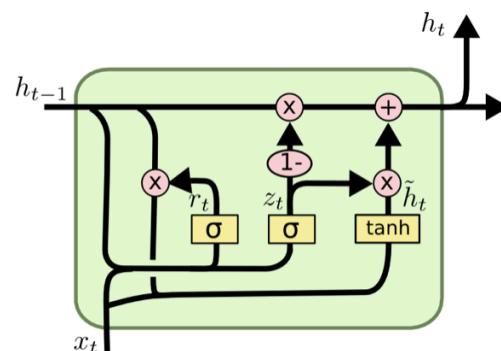
$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

**Coupled forget and input gates**



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

**Gated Recurrent Unit (GRU)**



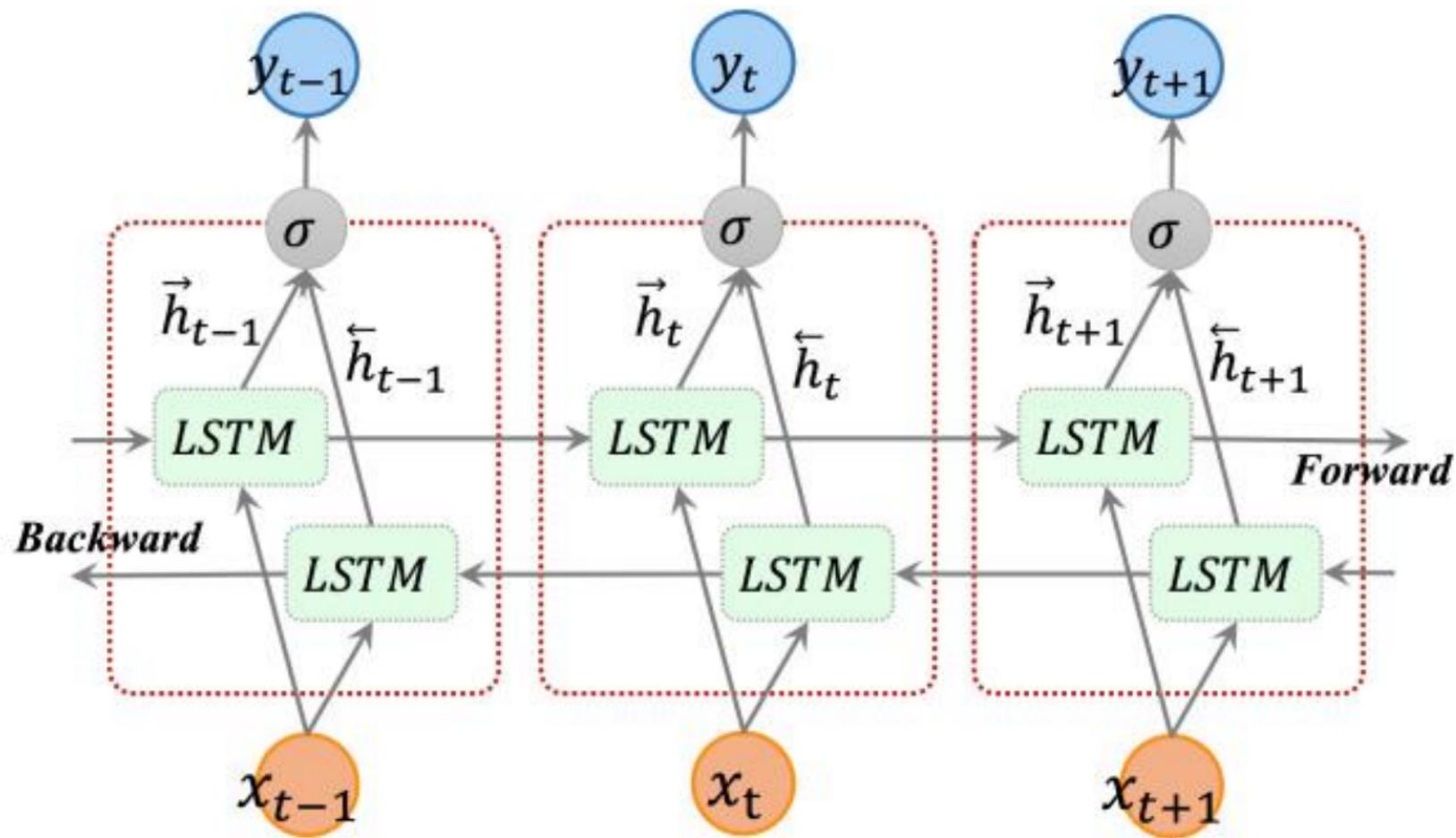
$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

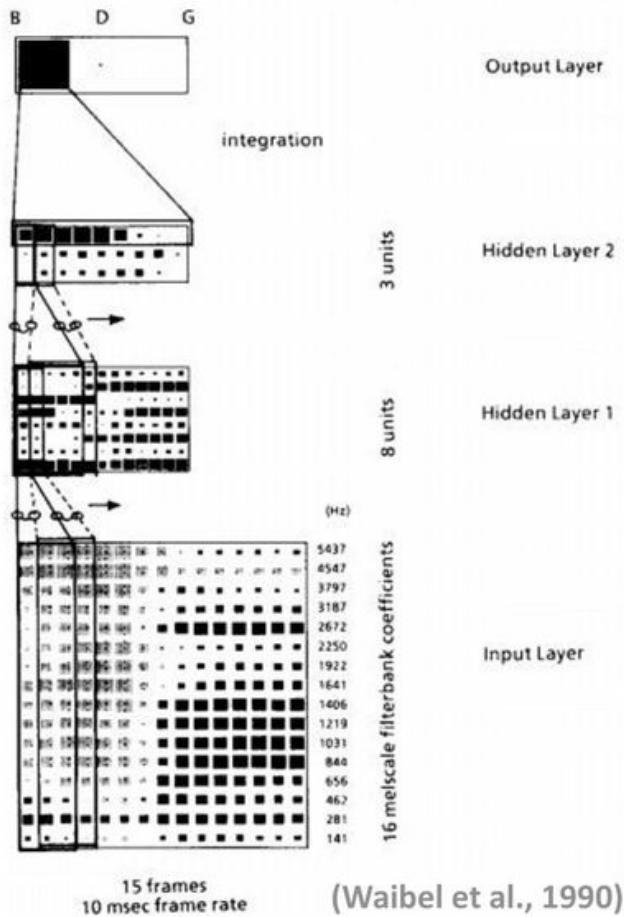
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

# Bidirectional RNN



**word2vec + LSTM =** ❤

# Time delay neural network

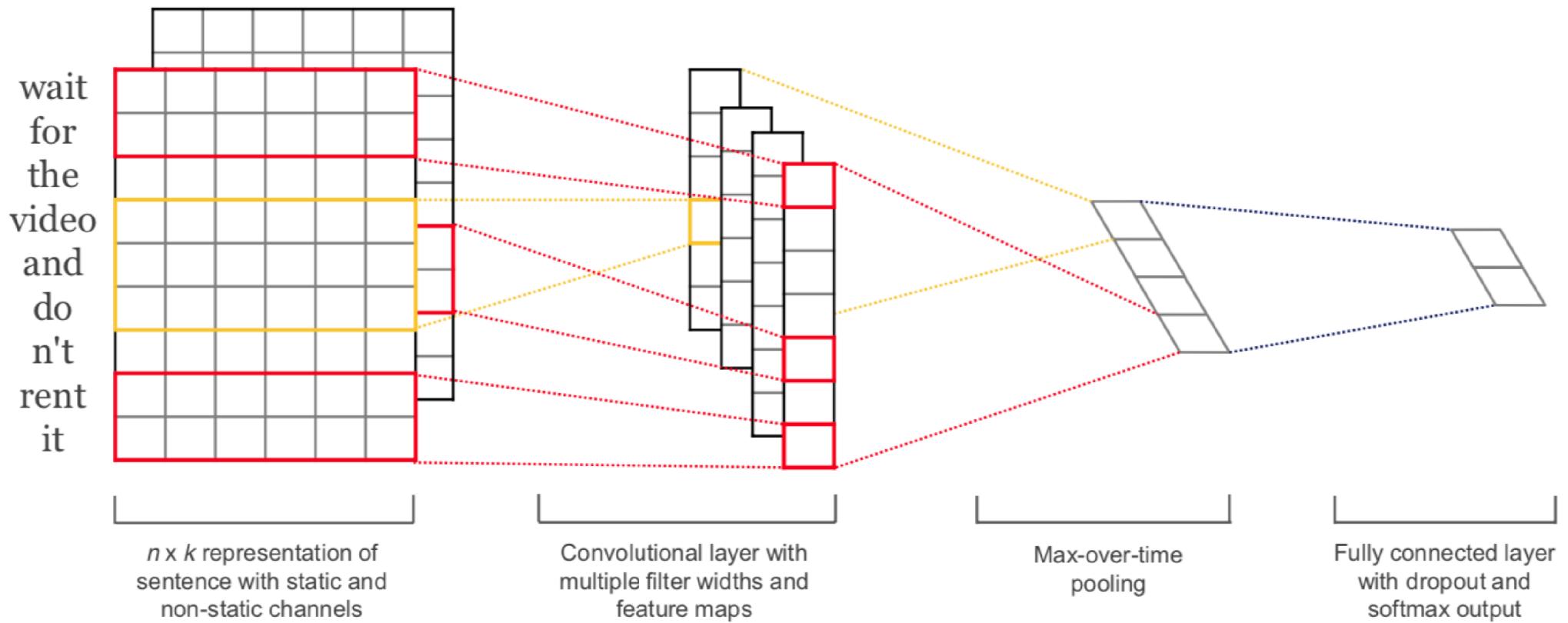


Waibel, Hinton, 1989

Phoneme Recognition Using Time-Delay Neural Networks



# CNN for texts (sequential)



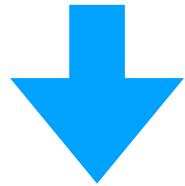
<https://habr.com/ru/company/ods/blog/353060/>

# Машинный перевод

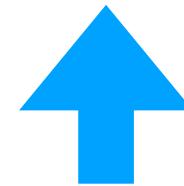


# Классический подход к переводу

*Je ne mangé pas six  
jours.*

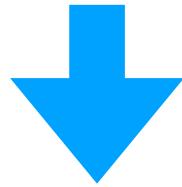


???



# Классический подход к переводу

*Je ne mangé pas six  
jours.*



*No como seis  
días.*



$$e_{best} = \operatorname{argmax}_e P(e | f)$$

# Классический подход к переводу

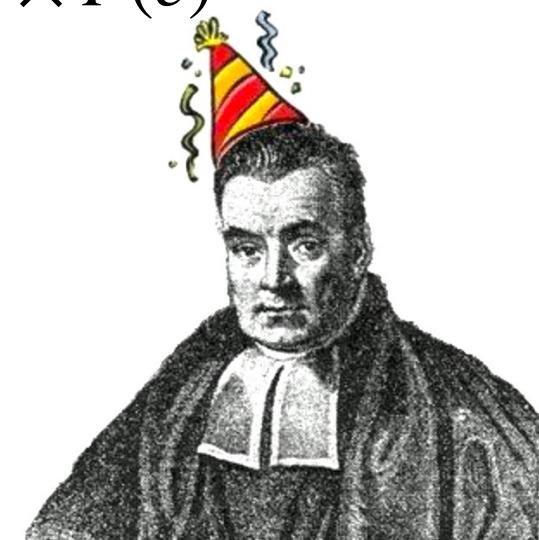
*Je ne mangé pas six  
jours.*



*No como seis  
días.*



$$\begin{aligned} e_{best} &= \operatorname{argmax}_e P(e | f) \\ &= \operatorname{argmax}_e P(f | e) \times P(e) \end{aligned}$$



# Классический подход к переводу

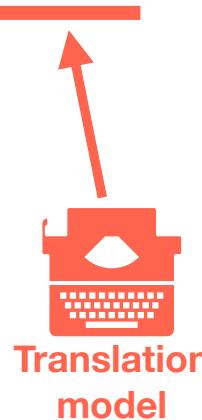
*Je ne mangé pas six  
jours.*



*No como seis  
días.*



$$\begin{aligned} e_{best} &= \operatorname{argmax}_e P(e | f) \\ &= \operatorname{argmax}_e \underline{P(f|e)} \times P(e) \end{aligned}$$



# Классический подход к переводу

*Je ne mangé pas six  
jours.*



*No como seis  
días.*

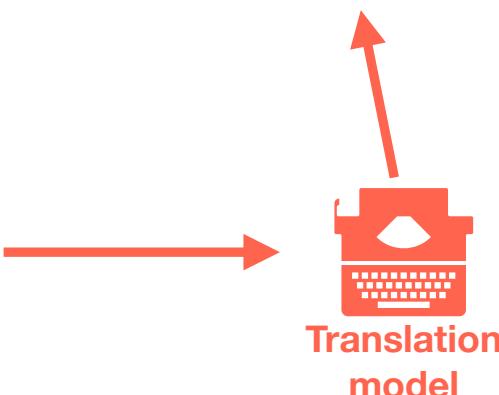


$$\begin{aligned} e_{best} &= \operatorname{argmax}_e P(e | f) \\ &= \operatorname{argmax}_e P(f | e) \times P(e) \end{aligned}$$

## *Word Alignment Matrix*

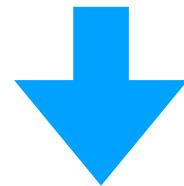
	Аппетит	приходит	во	время	еды	
The						
appetite						
comes						
with						
eating						

	1	2	3	4	5	6	7	
implemented								6
been								5
has								4
programme								3
the								2
And								1
Le								
	1	2	3	4	5	6	7	
application								
programme								
a								
été								
mis								
en								
the								



# Классический подход к переводу

*Je ne mangé pas six  
jours.*



*No como seis  
días.*



$$\begin{aligned} e_{best} &= \operatorname{argmax}_e P(e | f) \\ &= \operatorname{argmax}_e P(f | e) \times \underline{P(e)} \end{aligned}$$



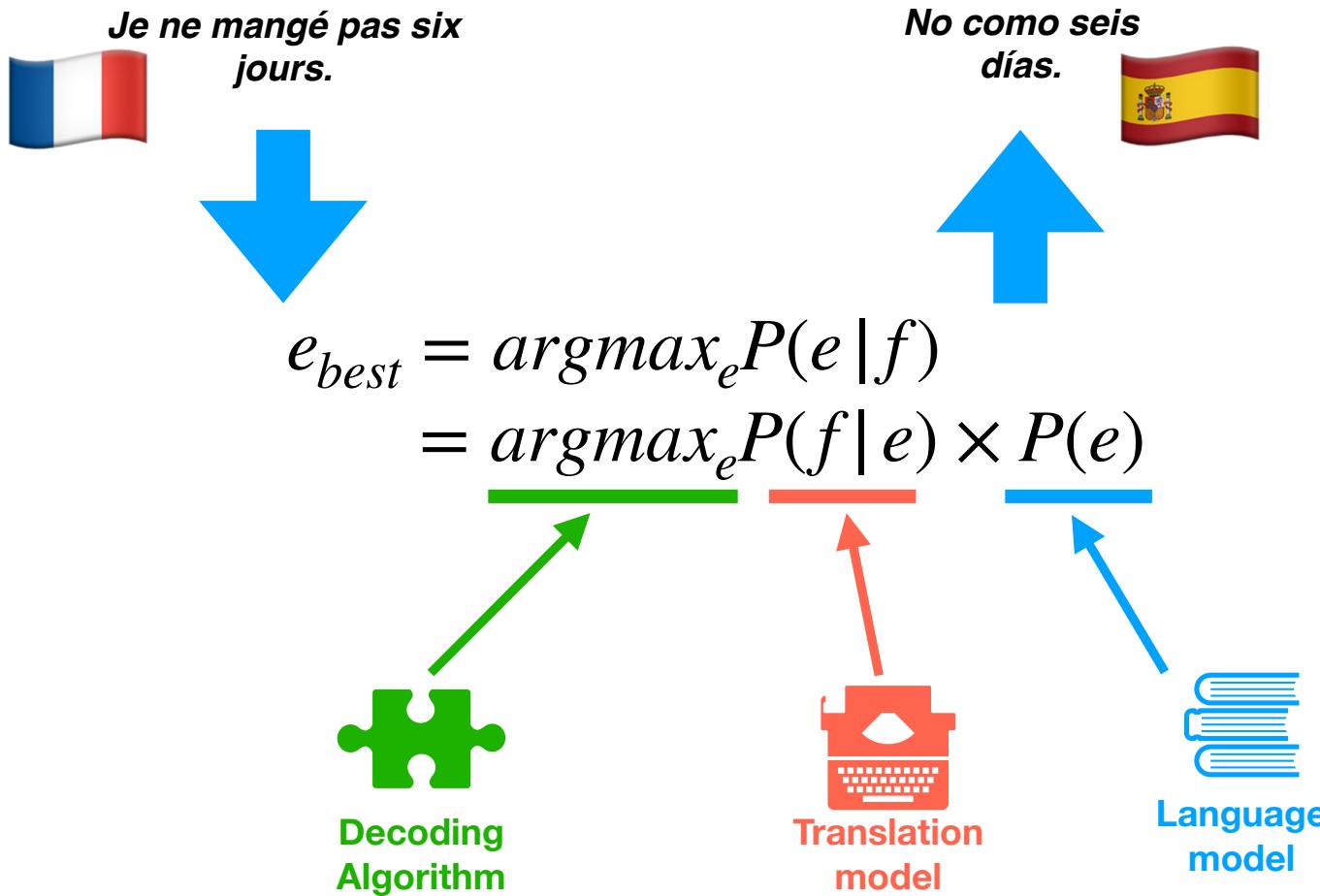
Translation  
model



Language  
model



# Классический подход к переводу



# Проблемы

*Je mangé*



*Como*

# Проблемы

*Je mangé*



*Como*

*Je ne mangé pas*



*No como*

# Проблемы

*Je mangé*



*Como*

*Je ne mangé pas*



*No como*

синонимы

идиомы

фразовые глаголы

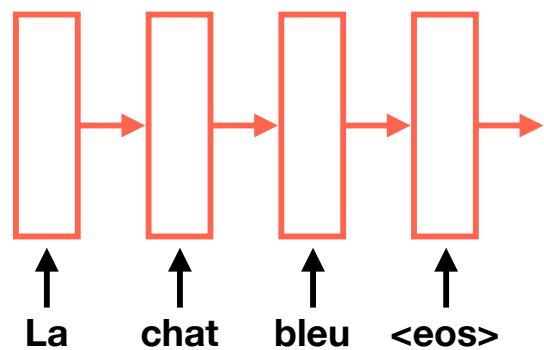
не формальная речь



# **Решение: sequence to sequence**

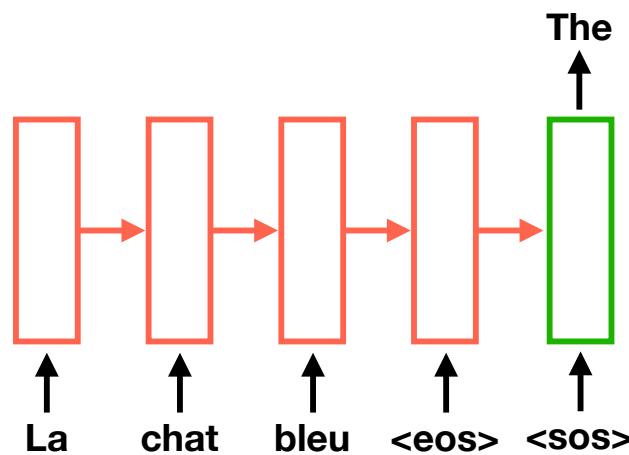
[Sequence to Sequence Learning with Neural Networks](#)

# Sequence to sequence



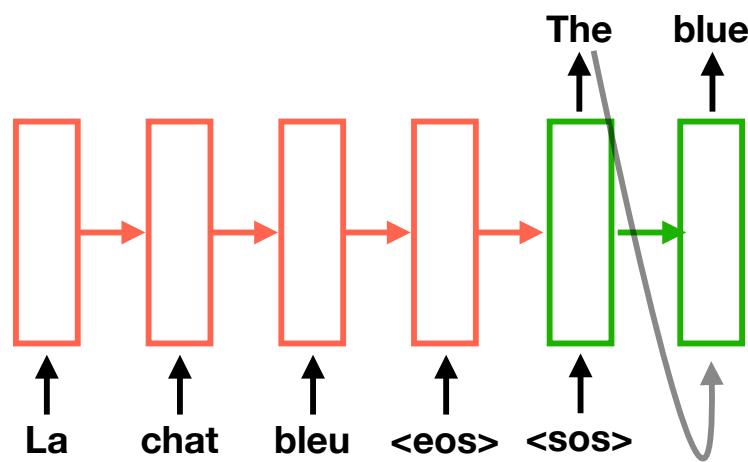
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence



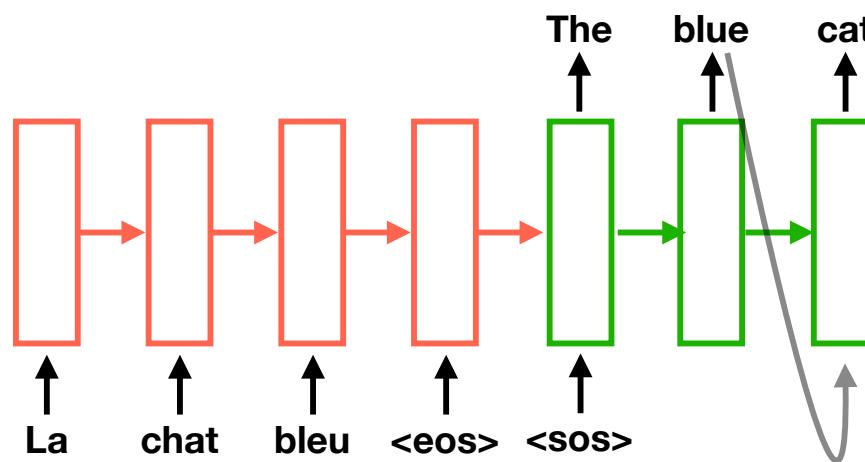
[Sequence to Sequence Learning with Neural Networks](#)

# Sequence to sequence



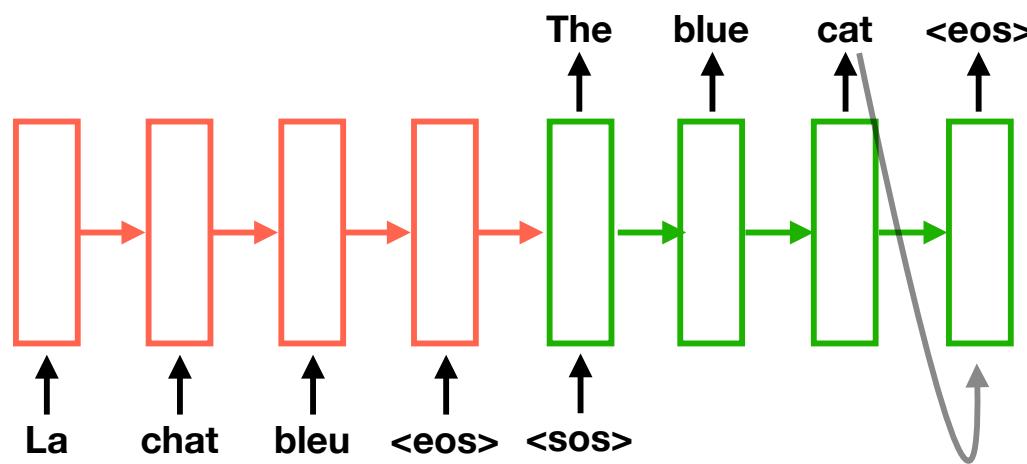
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence



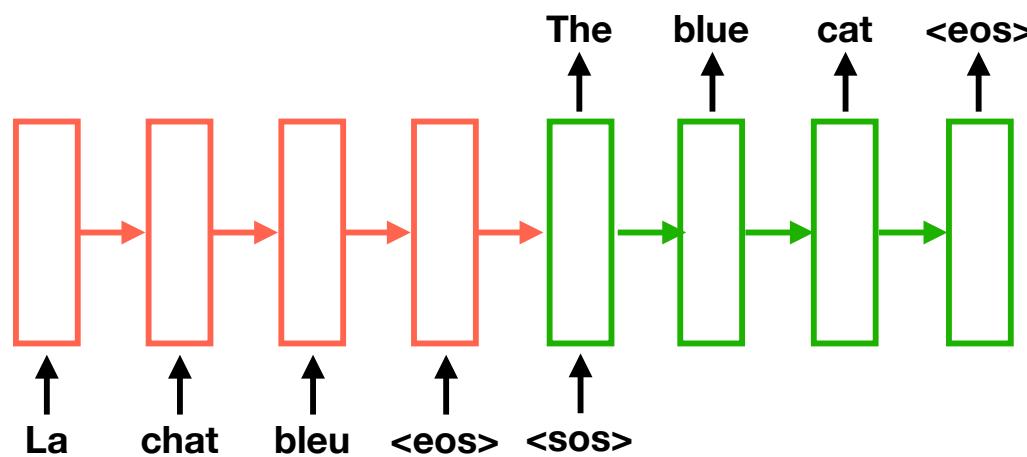
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence



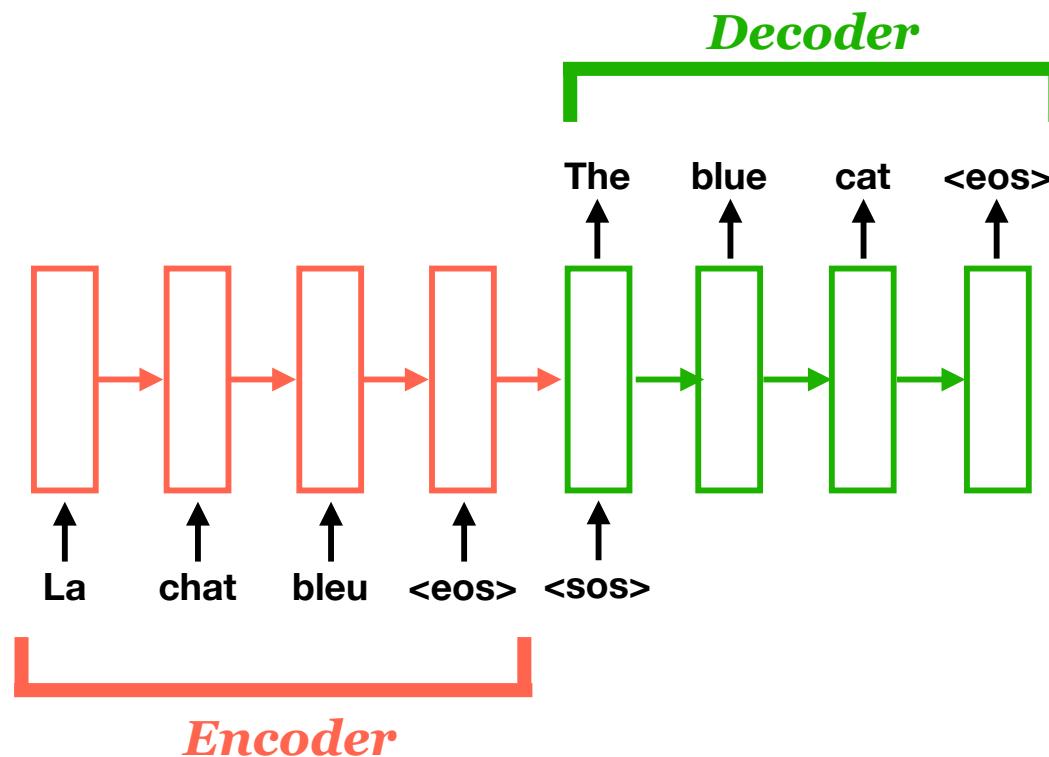
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence



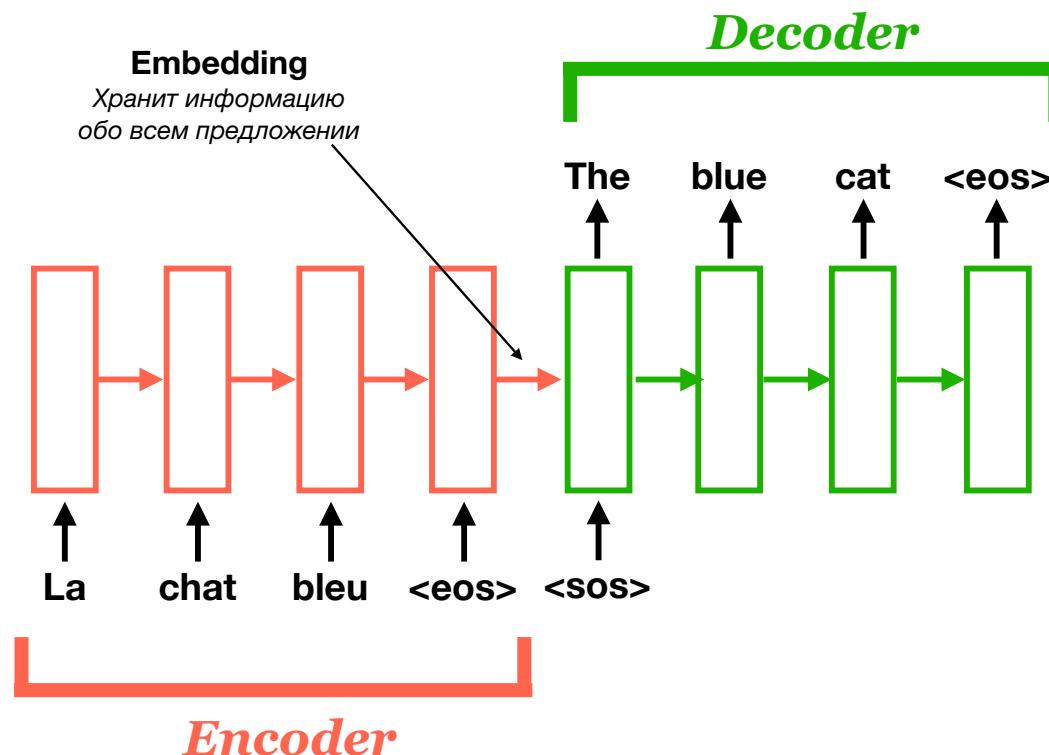
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence



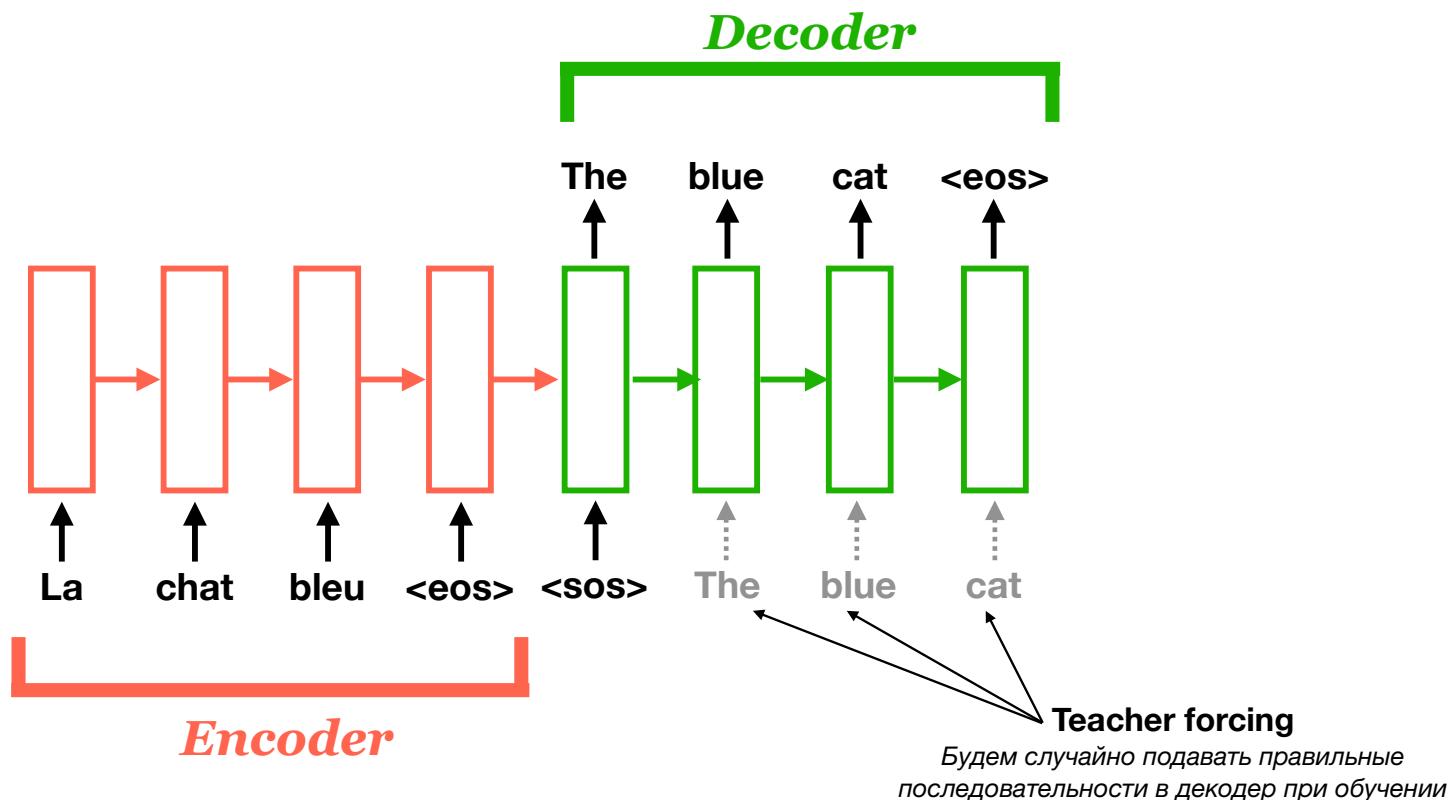
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence

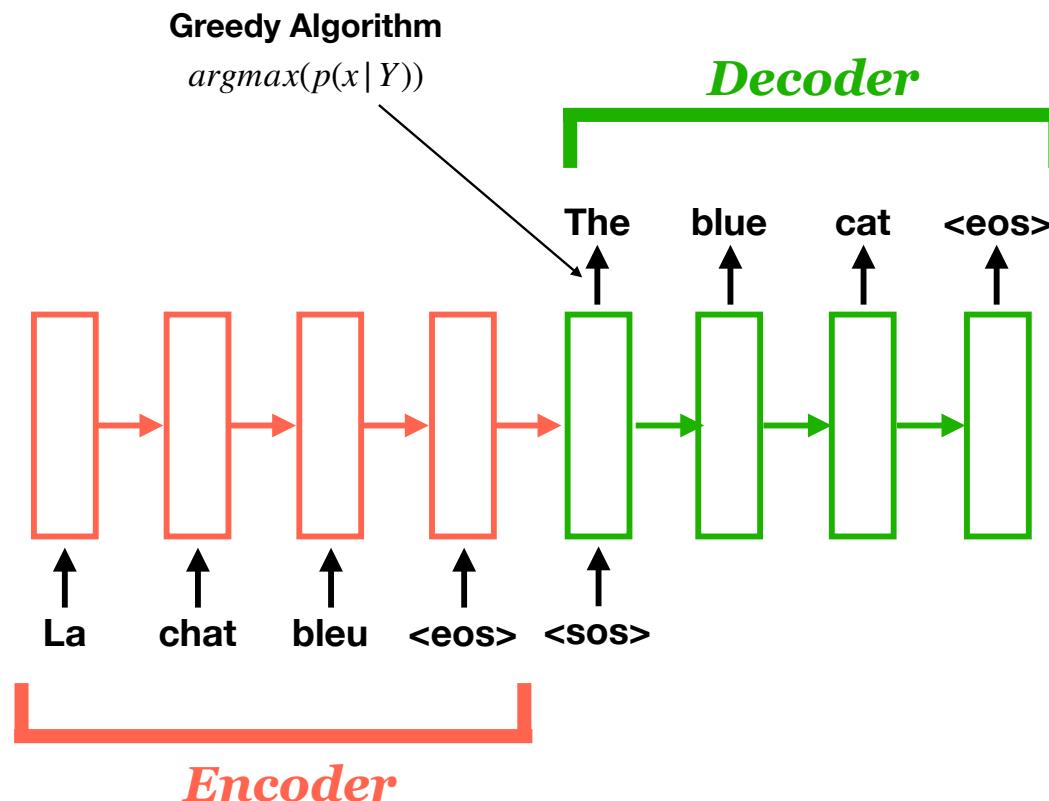


Sequence to Sequence Learning with Neural Networks

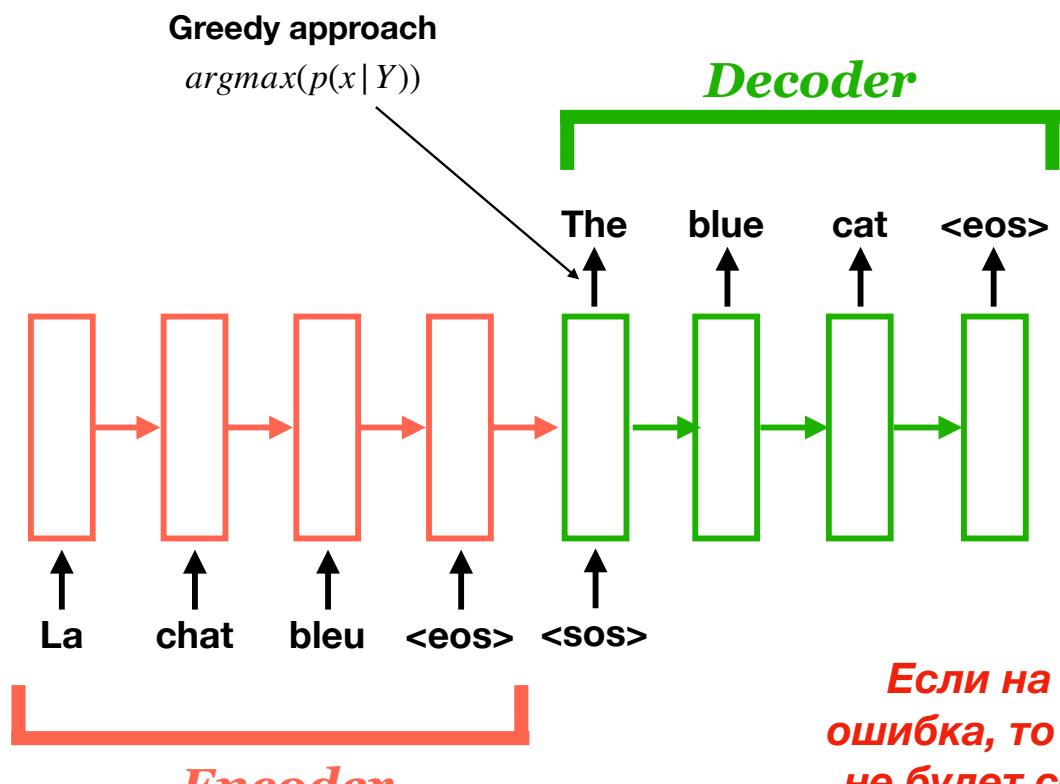
# Sequence to sequence



# Sequence to sequence

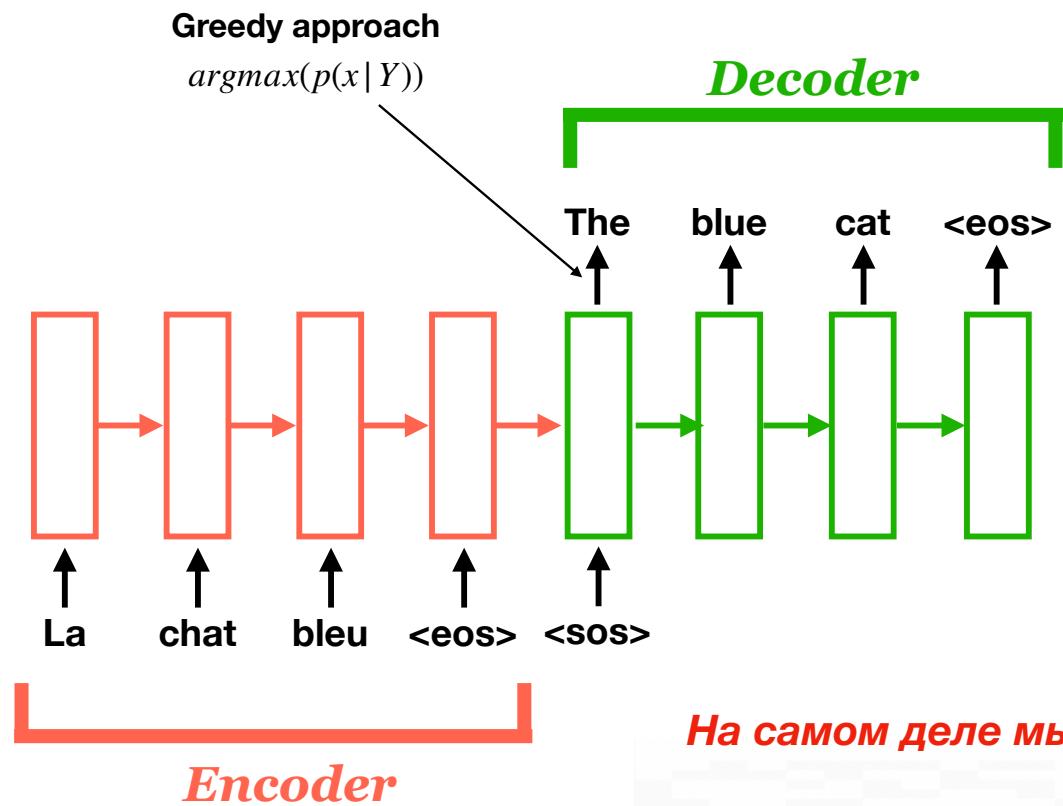


# Sequence to sequence



**Если на каком-то шаге будет ошибка, то дальнейшая генерация не будет соответствовать нашим ожиданиям**

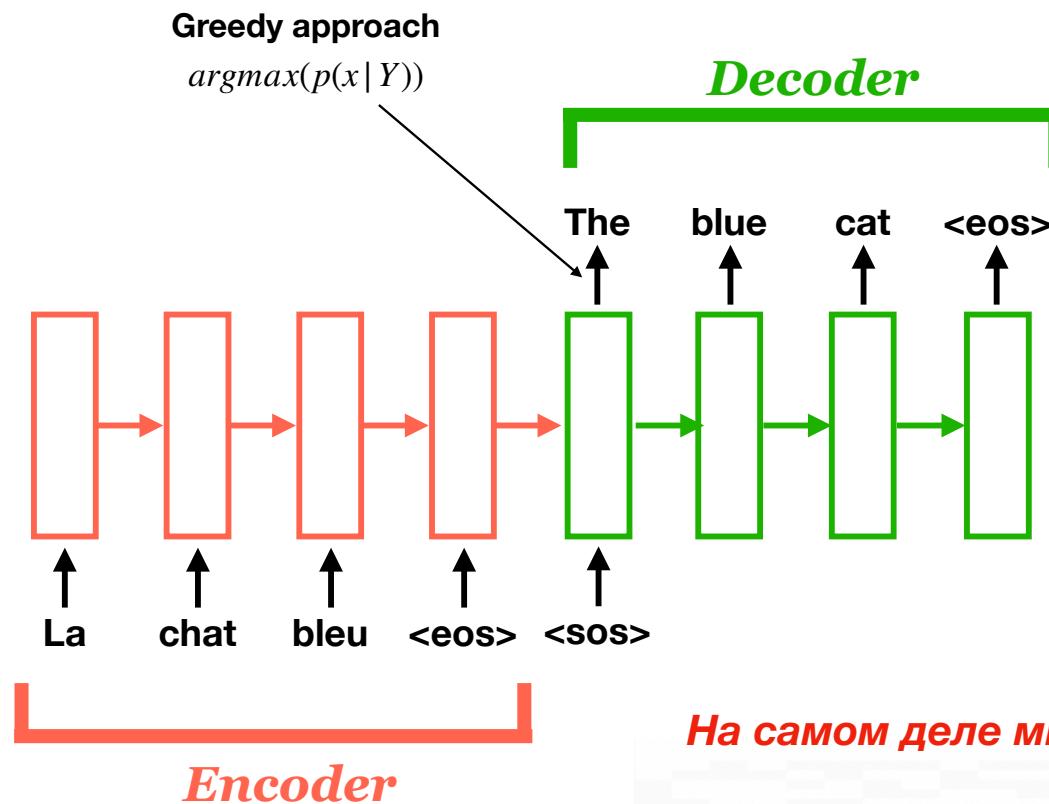
# Sequence to sequence



На самом деле мы хотим максимизировать:

$$P(y|x) = P(y_1|x) \prod_{t=2}^T P(y_t|y_1, \dots, y_{t-1}, x)$$

# Sequence to sequence

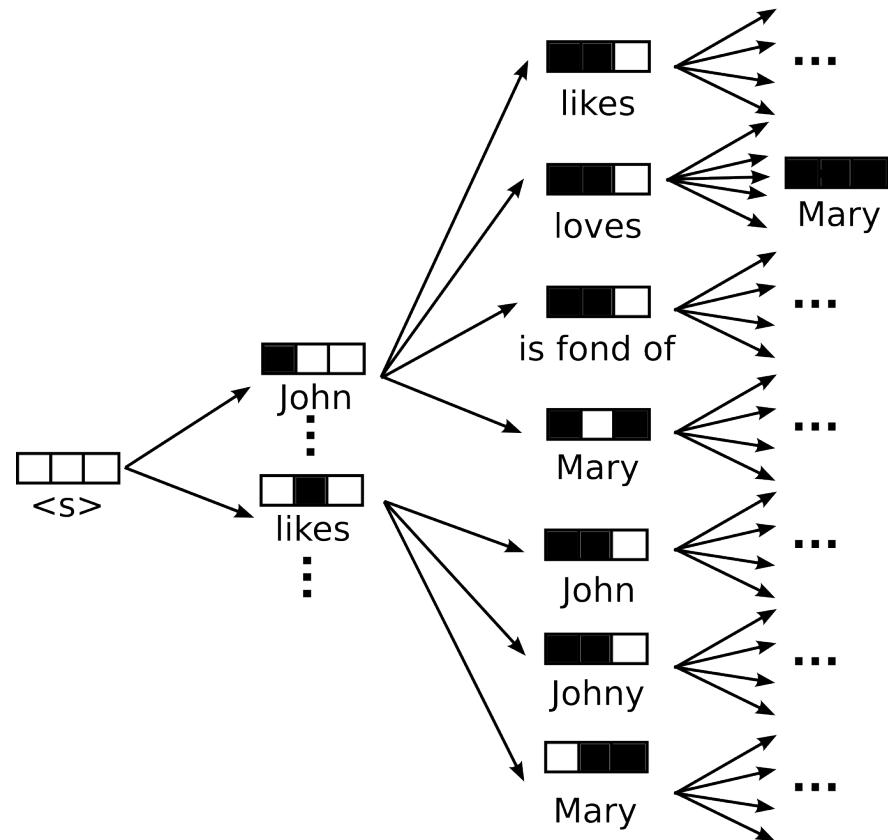


На самом деле мы хотим максимизировать:

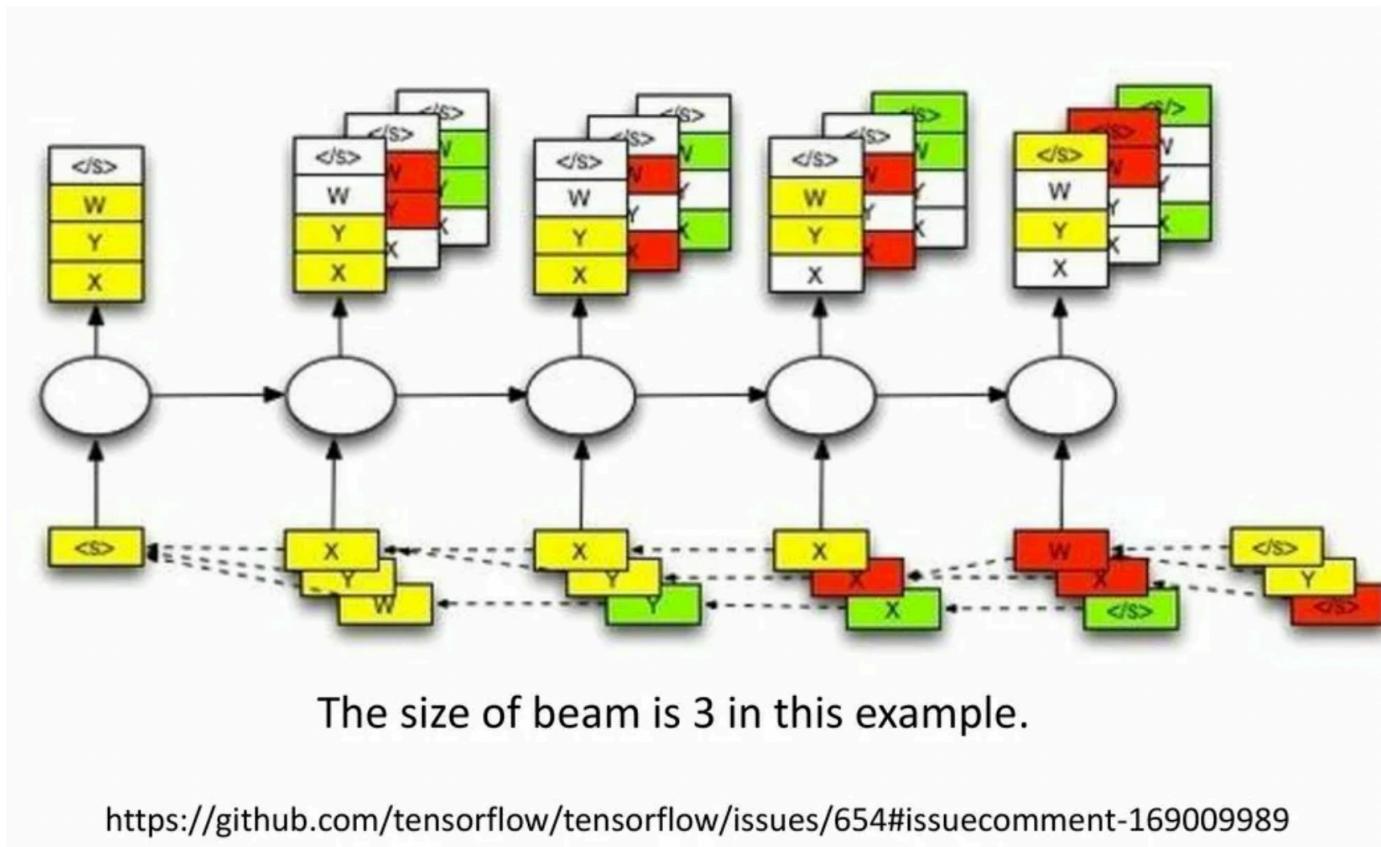
$$P(y|x) = P(y_1|x) \prod_{t=2}^T P(y_t|y_1, \dots, y_{t-1}, x)$$

Но если считать для всех возможных, то возникает экспоненциальная сложность

# Beam search



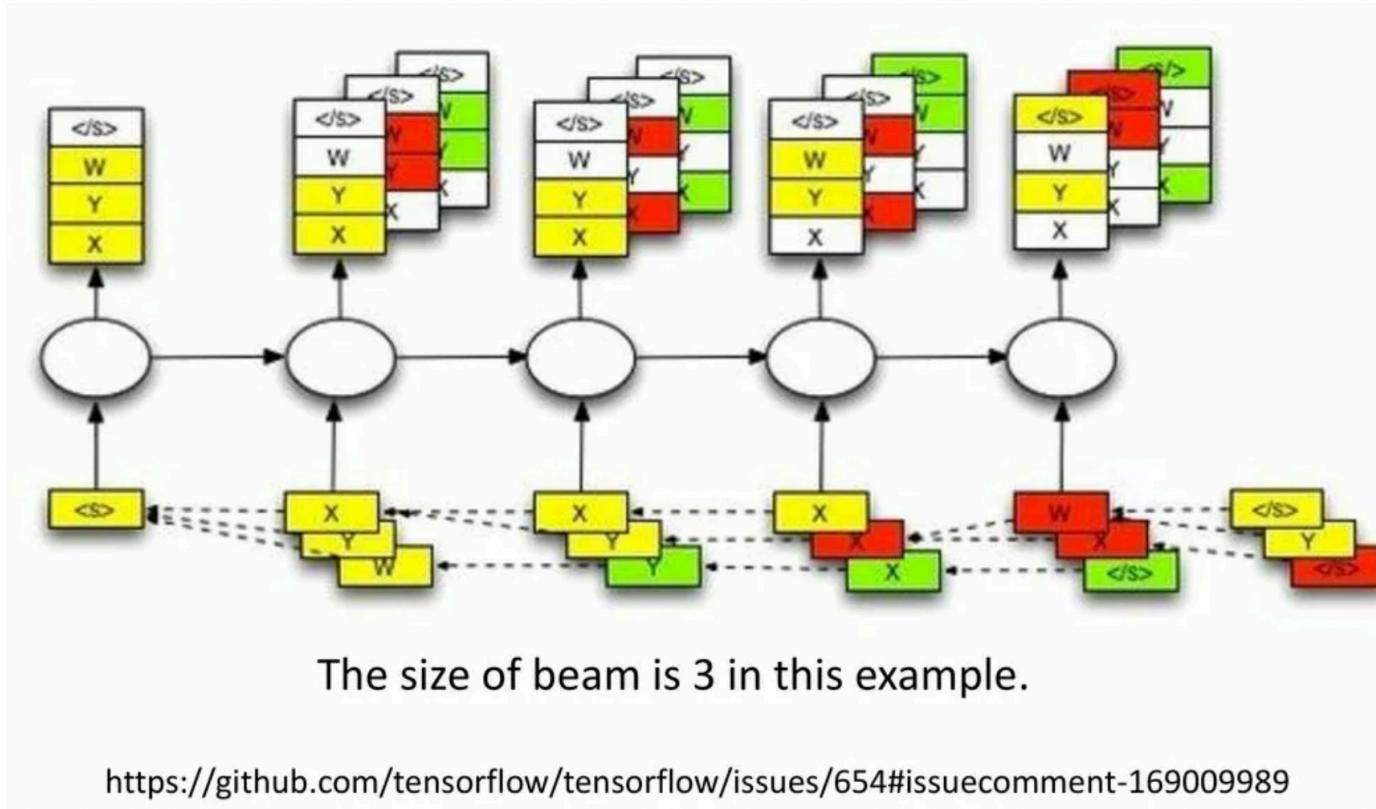
# Sequence to sequence



Sequence to Sequence Learning with Neural Networks

# Sequence to sequence

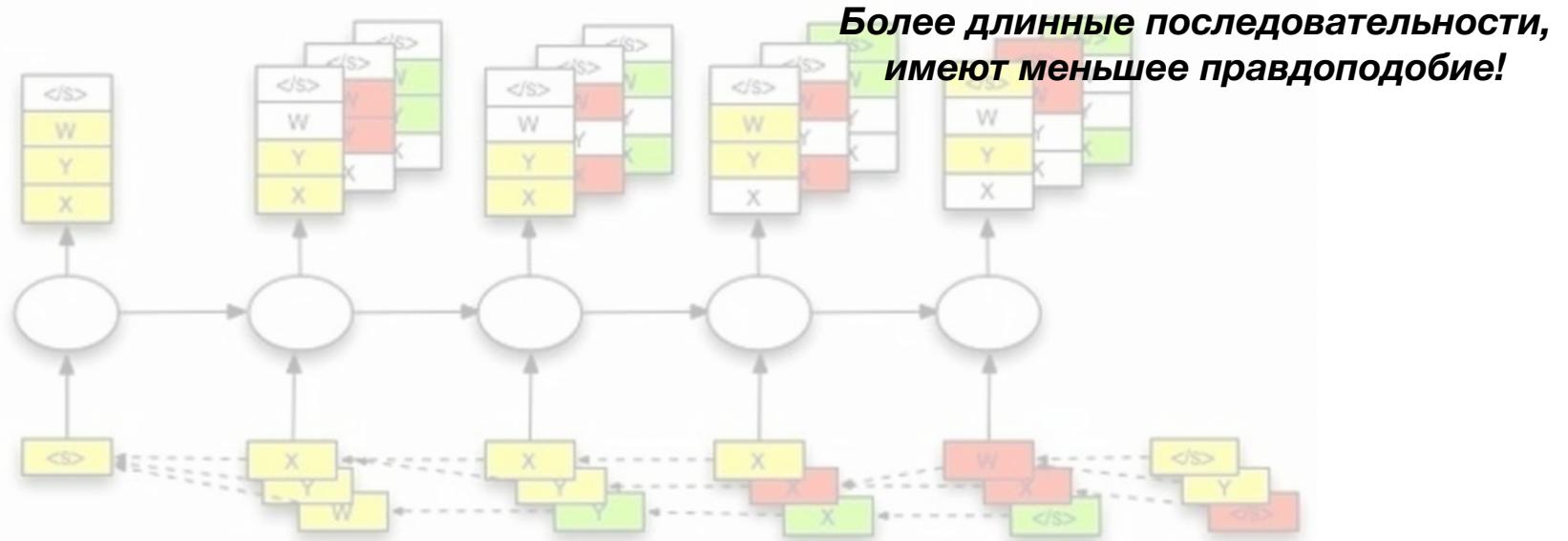
$$\text{score}(y_1, \dots, y_t) = \log P_{\text{LM}}(y_1, \dots, y_t | x) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$



Sequence to Sequence Learning with Neural Networks

# Sequence to sequence

$$\text{score}(y_1, \dots, y_t) = \log P_{\text{LM}}(y_1, \dots, y_t | x) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$



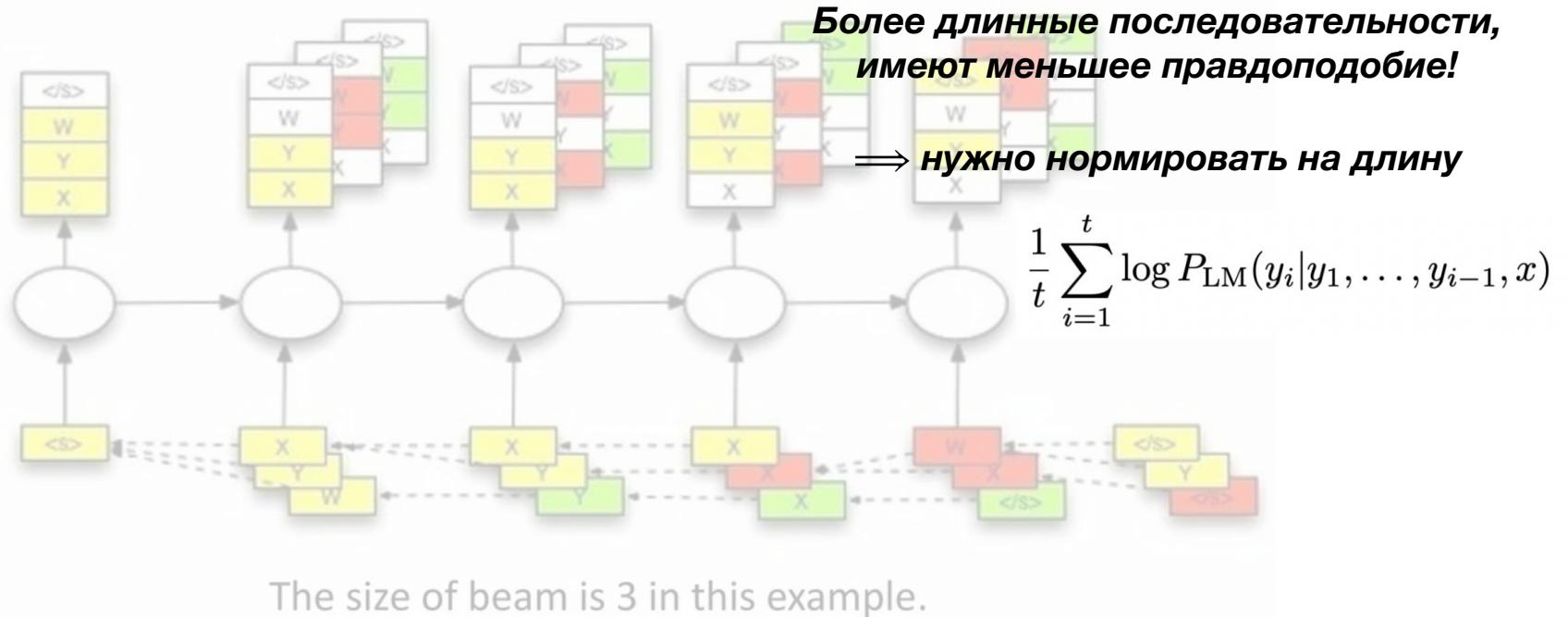
The size of beam is 3 in this example.

<https://github.com/tensorflow/tensorflow/issues/654#issuecomment-169009989>

Sequence to Sequence Learning with Neural Networks

# Sequence to sequence

$$\text{score}(y_1, \dots, y_t) = \log P_{\text{LM}}(y_1, \dots, y_t | x) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$$

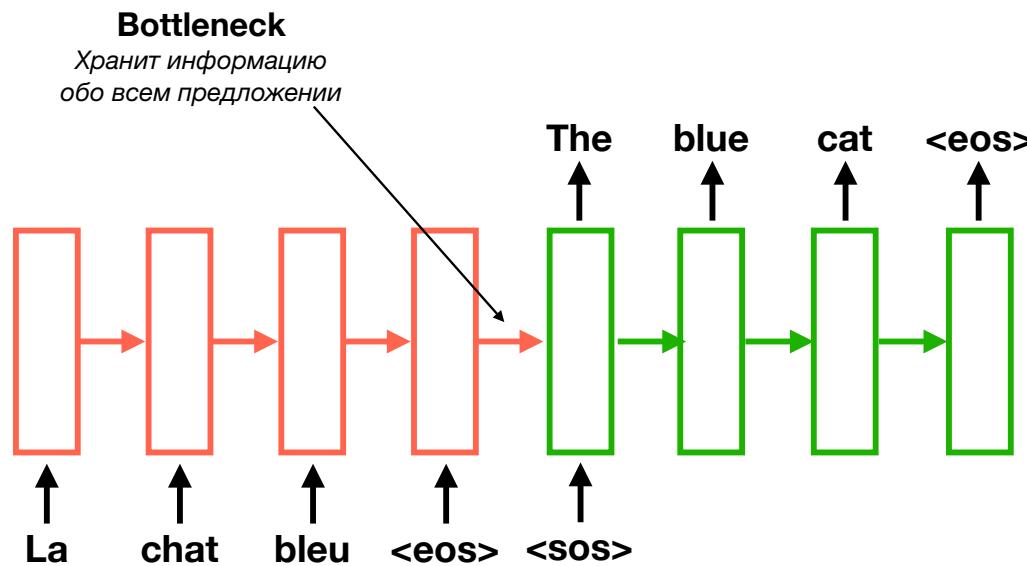


The size of beam is 3 in this example.

<https://github.com/tensorflow/tensorflow/issues/654#issuecomment-169009989>

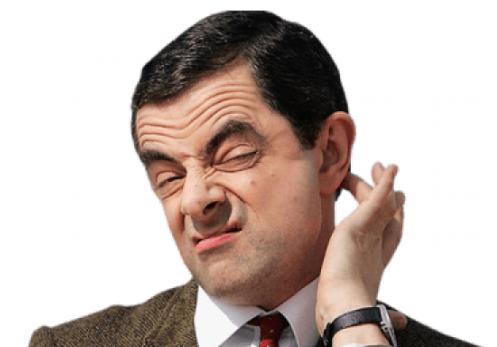
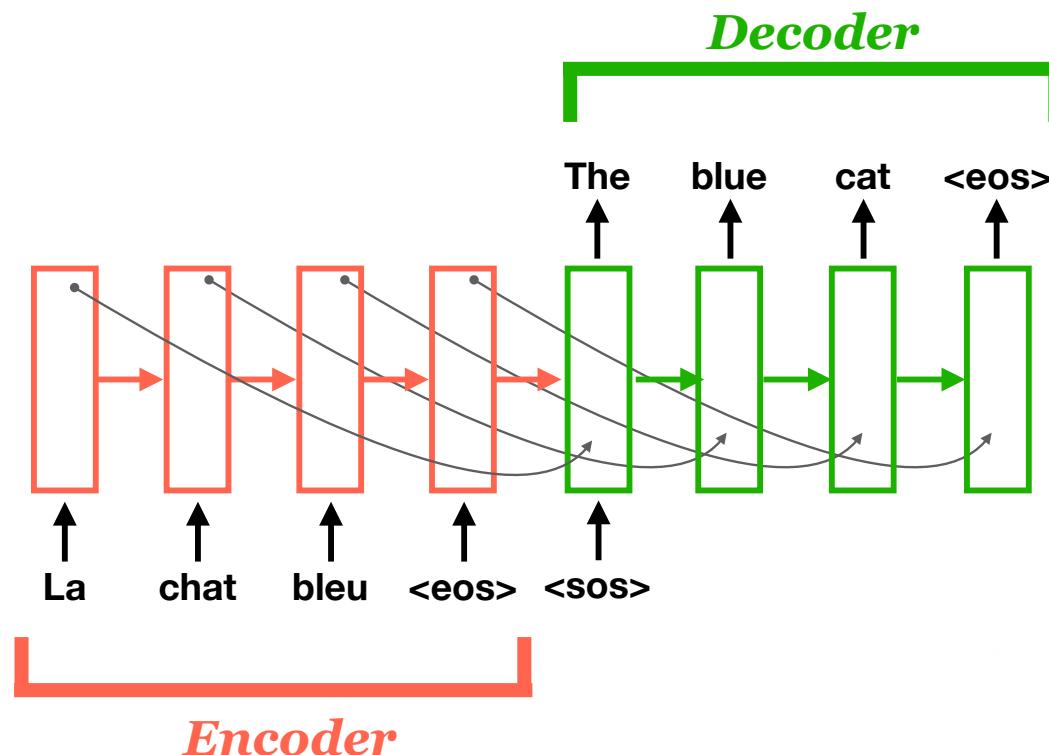
Sequence to Sequence Learning with Neural Networks

# Sequence to sequence



[Sequence to Sequence Learning with Neural Networks](#)

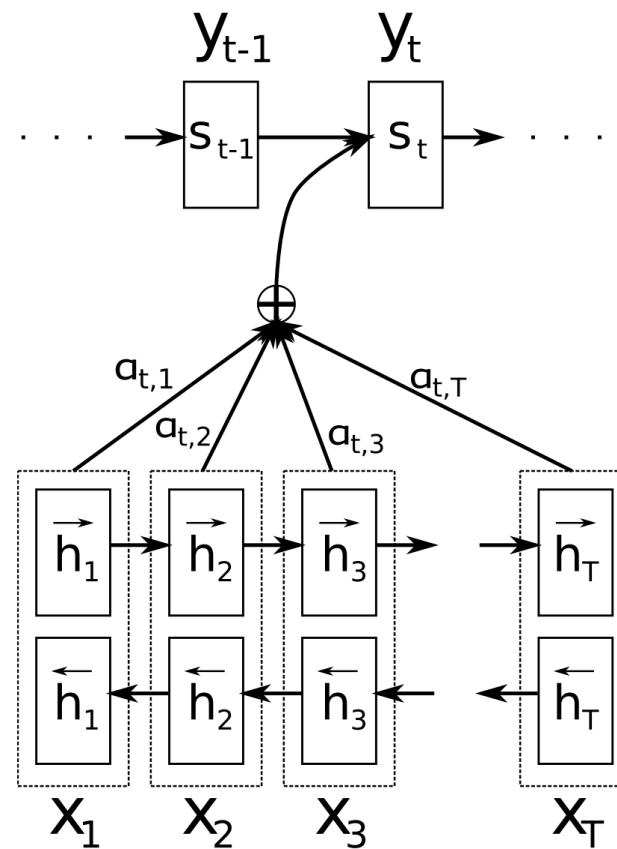
# Sequence to sequence



Sequence to Sequence Learning with Neural Networks

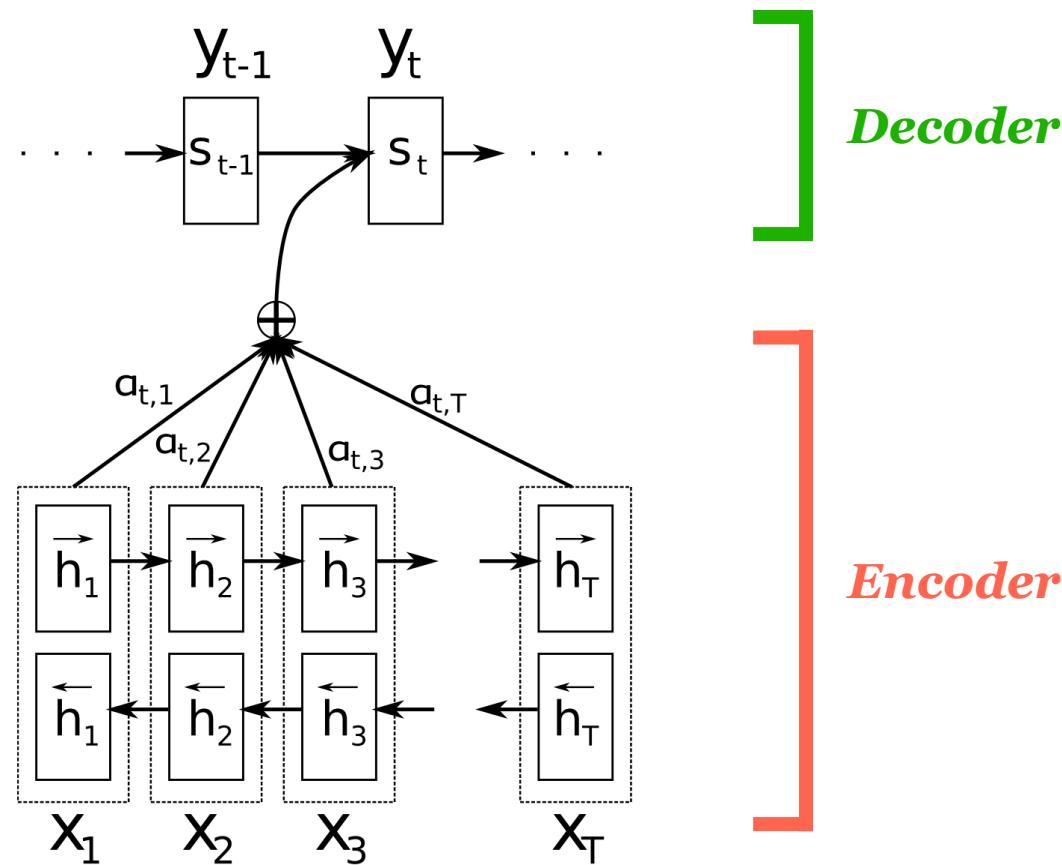
**Attention Is All You Need**

# Attention



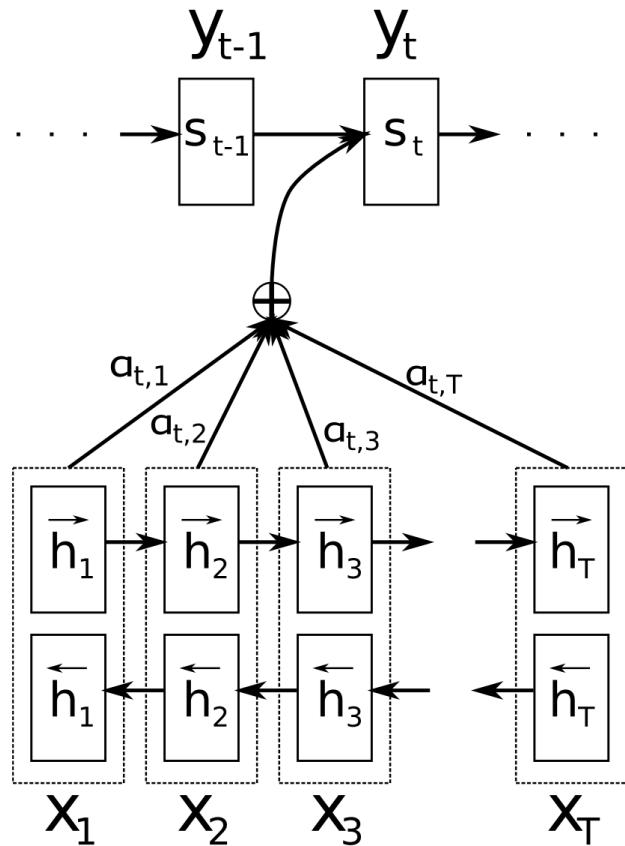
Neural Machine Translation by Jointly Learning to Align and Translate

# Attention



Neural Machine Translation by Jointly Learning to Align and Translate

# Attention



Скрытый слой RNN

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

Взвешенная сумма

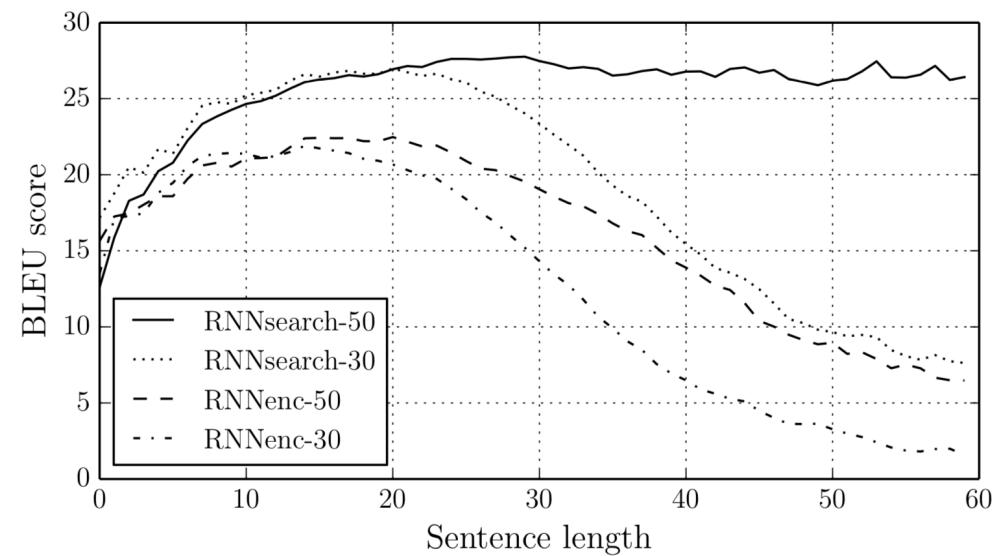
$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1}, h_j)$$

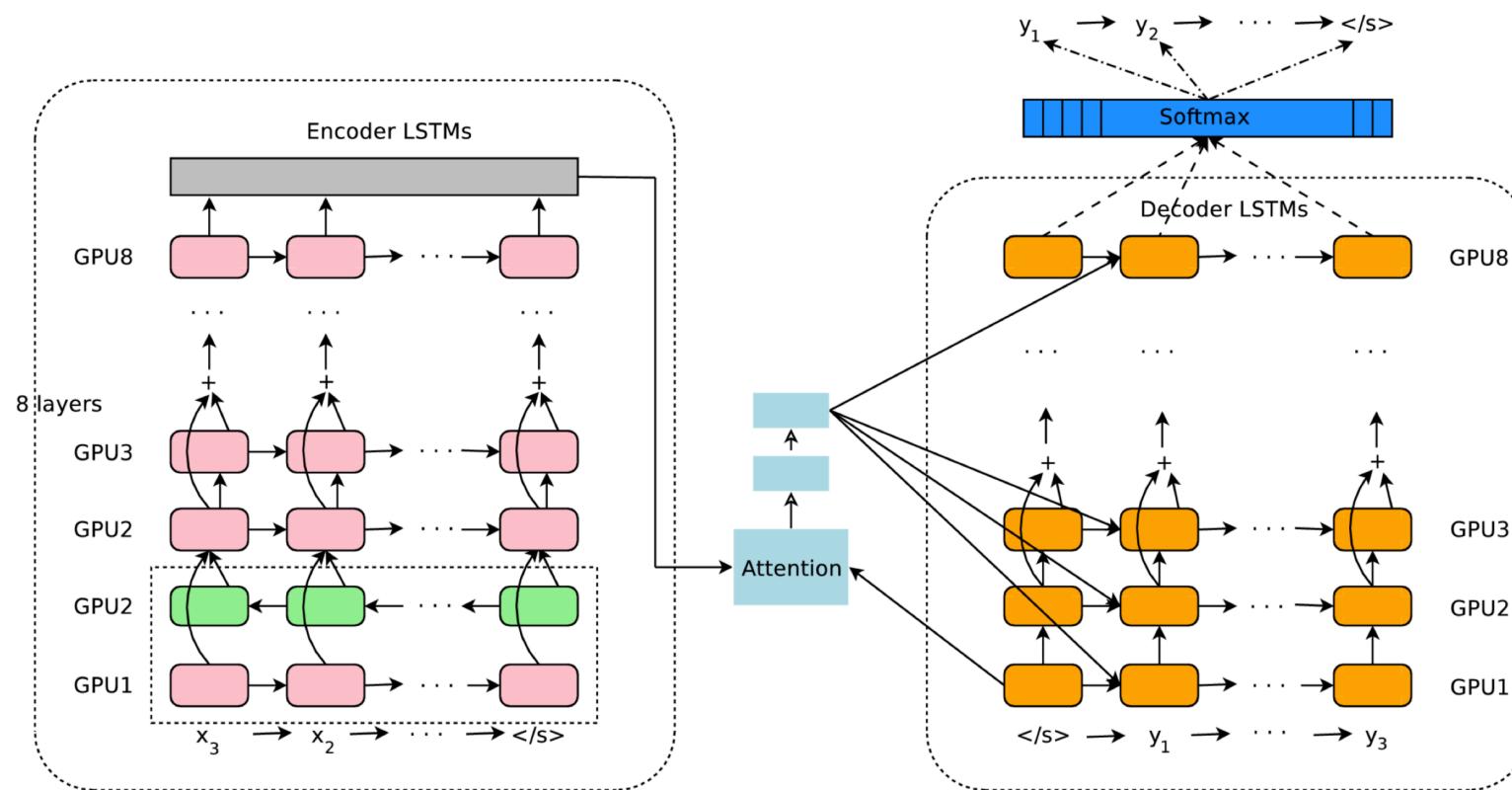
Neural Machine Translation by Jointly Learning to Align and Translate

# Attention



Neural Machine Translation by Jointly Learning to Align and Translate

# Attention



Bridging the Gap between Human and Machine Translation