

# Глубокое обучение и вообще

Ульянкин Филипп

1 июня 2022 г.

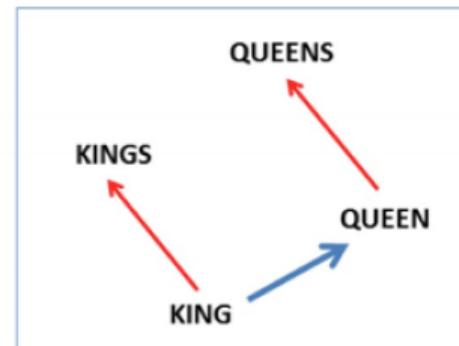
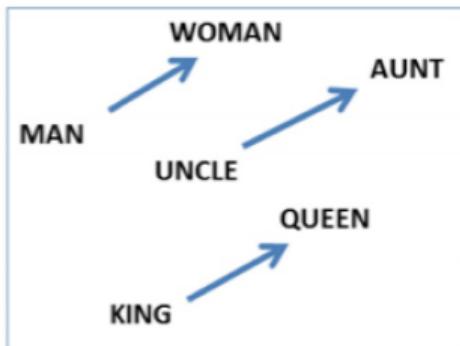
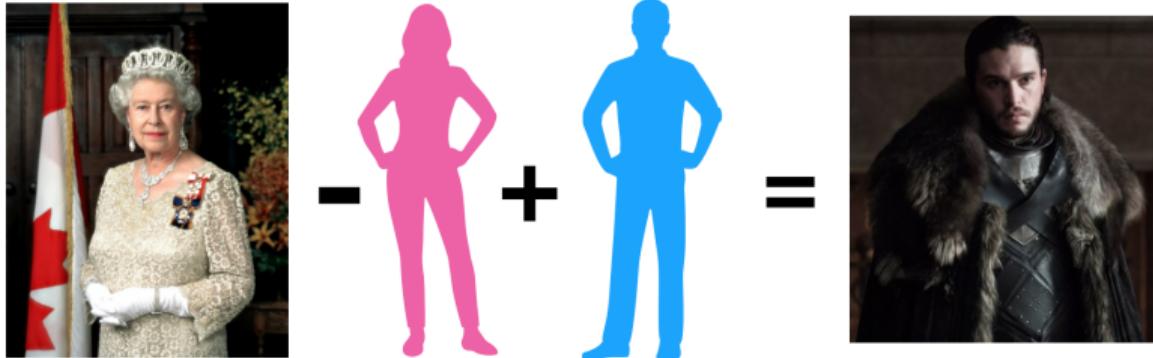
**Посиделка 12:** Seq2Seq и Attention

# Agenda

- Мультимодальные модели
- История автоперевода
- Нейросетевой автоперевод
- Механизмы внимания

# Мультимодальные модели

# Word2vec: семантическое пространство



(Mikolov et al., NAACL HLT, 2013)

# Векторные пространства

- Мы умеем обучать векторные пространства для какого-то одного типа объектов
- Вот бы можно было учить векторные пространства сразу для разных объектов

# Мультимодальное векторное пространство (2014)

Nearest images



- dog + cat =



- cat + dog =



- plane + bird =



- man + woman =



(a) Simple cases

# Мультимодальное векторное пространство (2014)

Nearest images



- blue + red =



- blue + yellow =



- yellow + red =



- white + red =



(b) Colors

# Мультимодальное векторное пространство (2014)

Nearest images



- day + night =



- flying + sailing =



- bowl + box =



- box + bowl =



(c) Image structure

# Мультимодальное векторное пространство (2014)

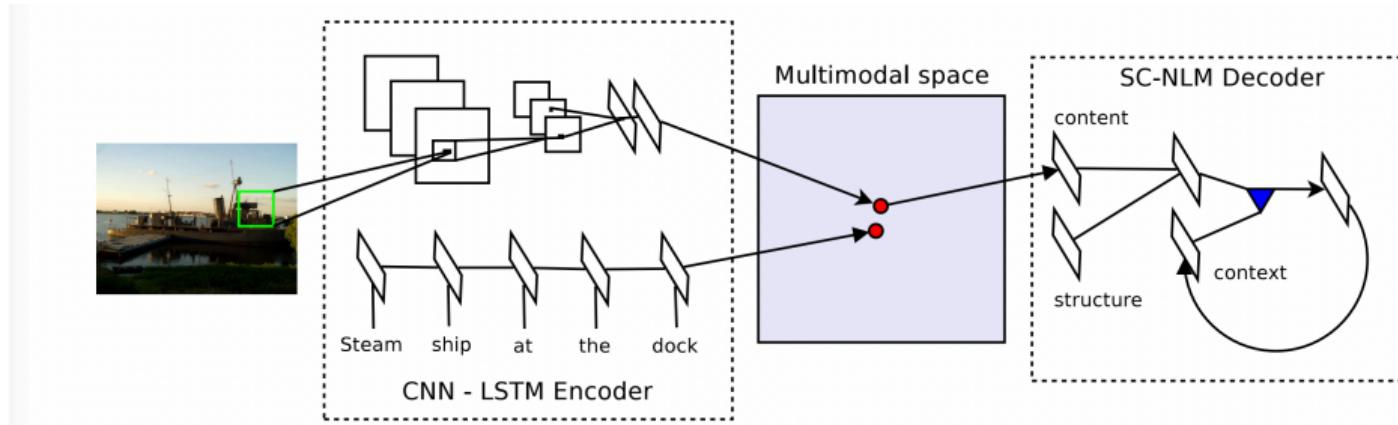
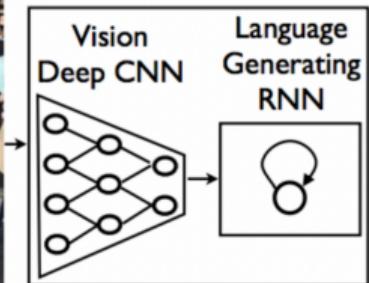


Figure 2: **Encoder:** A deep convolutional network (CNN) and long short-term memory recurrent network (LSTM) for learning a joint image-sentence embedding. **Decoder:** A new neural language model that combines structure and content vectors for generating words one at a time in sequence.

<https://arxiv.org/pdf/1411.2539.pdf>

# Генерация текста по картинке (2015)

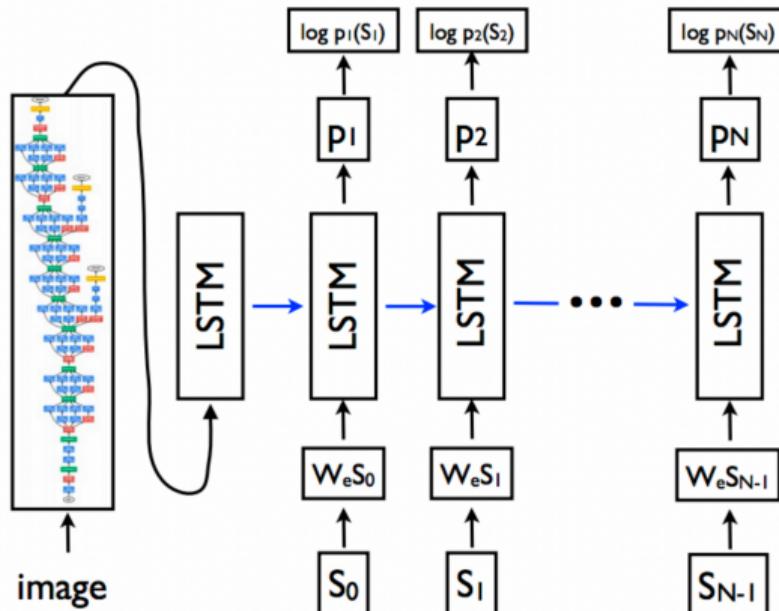


**A group of people shopping at an outdoor market.**

**There are many vegetables at the fruit stand.**

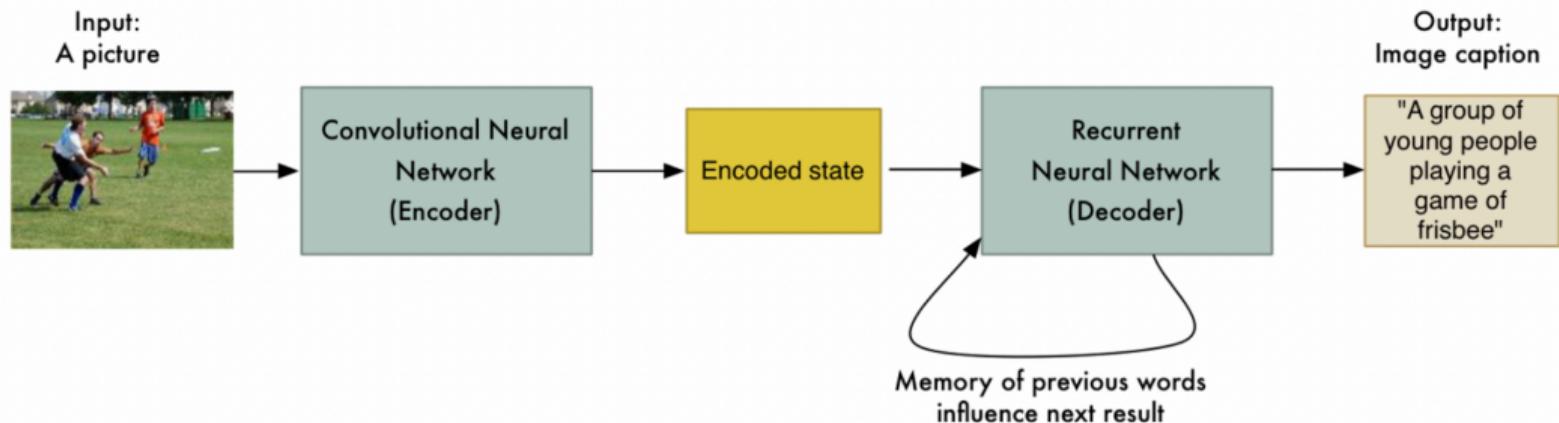
<https://arxiv.org/pdf/1411.4555.pdf>

# Генерация текста по картинке (2015)



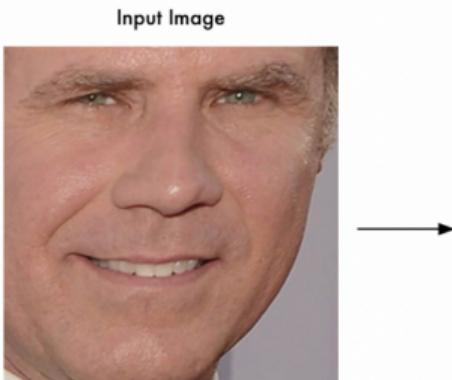
<https://arxiv.org/pdf/1411.4555.pdf>

# Генерация текста по картинке (2015)



<https://arxiv.org/pdf/1411.4555.pdf>

# Генерация текста по картинке (2015)



128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567115924778	-0.10568545013666	-0.12728653848171	-0.076288616525173
-0.061418771743774	-0.074287034571171	-0.06536532527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.00410191227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.1221668576002	-0.00727775558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.1340749858099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

<https://arxiv.org/pdf/1411.4555.pdf>

# Генерация текста по картинке (2015)

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



Describes without errors

Describes with minor errors

Somewhat related to the image

Unrelated to the image

Figure 5. A selection of evaluation results, grouped by human rating.

# Генерация текста по картинке (2015)



a bunch of bananas are hanging from a ceiling

<https://arxiv.org/pdf/1411.4555.pdf>

# Генерация картинок по тексту (2022)

DALL-E

Edit the detailed description

Surprise me Upload

the painting American Gothic, with two dogs holding pepperoni pizza instead of the farmers holding a pitchfork →

Report issue ↗



<https://openai.com/dall-e-2/>

<https://arxiv.org/pdf/2204.06125.pdf>

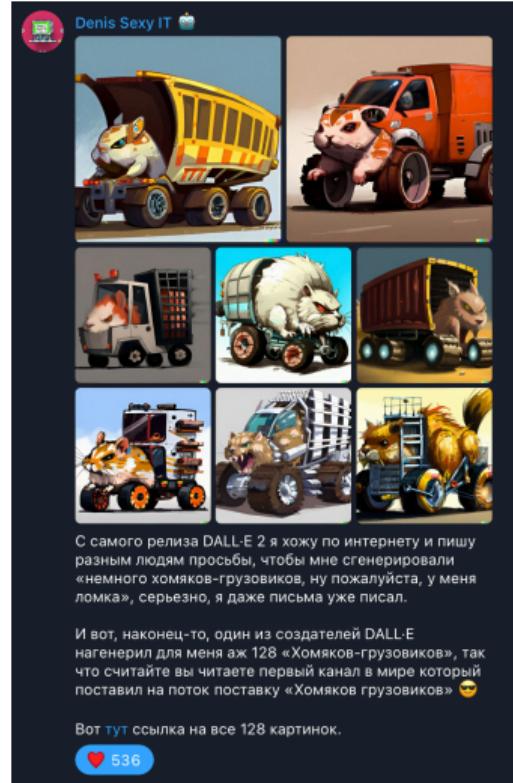
# Генерация картинок по тексту (2022)



<https://openai.com/dall-e-2/>

<https://arxiv.org/pdf/2204.06125.pdf>

# Генерация картинок по тексту (2022)



С самого релиза DALL-E 2 я хожу по интернету и пишу разным людям просьбы, чтобы мне сгенерировали «немного хомяков-грузовиков, ну пожалуйста, у меня ломка», серьезно, я даже письма уже писал.

И вот, наконец-то, один из создателей DALL-E нагенерил для меня аж 128 «Хомяков-грузовиков», так что считайте вы читаете первый канал в мире который поставил на поток поставку «Хомяков грузовиков» 😎

Вот [тут](#) ссылка на все 128 картинок.

536

<https://t.me/denissexy>

# Генерация картинок по тексту (2022)

A blue jay standing on a large basket of rainbow macarons.



# Imagen

unprecedented photorealism × deep level of language understanding

Google Research, Brain Team

<https://imagen.research.google/>

# История автоперевода

## Переводчики

The screenshot shows the Yandex.Translate website. On the left, there's a toolbar with icons for text input, microphone, clipboard, and font size. Below it is a text input field with placeholder text "Введите текст или адрес сайта". To its right are two language selection boxes: one for "АНГЛИЙСКИЙ" (selected) and one for "РУССКИЙ". Between them is a double-headed arrow icon. On the far right, there's another toolbar with icons for bookmark, microphone, clipboard, and other sharing options.

The image shows the Google Translate web interface. At the top, there's a navigation bar with the Google logo and a menu icon. Below it, two tabs are visible: 'Текст' (Text) which is selected and highlighted in blue, and 'Документы' (Documents). The main workspace has a header with language selection: 'ОПРЕДЕЛИТЬ ЯЗЫК' (Detect Language), 'АНГЛИЙСКИЙ' (English), 'РУССКИЙ' (Russian), 'НЕМЕЦКИЙ' (German), and dropdown arrows. To the right of this header, another row of languages is shown: 'РУССКИЙ' (Russian), 'АНГЛИЙСКИЙ' (English), 'УКРАИНСКИЙ' (Ukrainian), and a dropdown arrow. Below the header, there's a large input area with a microphone icon and a character count of '0/5000'. To the right of the input area, the word 'Перевод' (Translation) is displayed in the output field. At the bottom right of the page, there's a link 'Отправить отзыв' (Send feedback).

# Кто и когда захотел

- Холодная война, 1954 год. США и СССР хотят получить машину для автопереводов
- Джорджтаунский эксперимент (IBM): перевод 60 предложений с перфокарт
- В СССР аналогичный эксперимент
- В обоих государствах все предложения для перевода тщательно подобраны и оттестированы :(

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

# Кто и когда захотел

- Учёные обещают, что в течение ближайших 5 лет задача машинного перевода будет решена.
- Через 12 лет, в 1966 американский комитет ALPAC публикует отчёт, в котором называет машинный перевод дорогим, неточным и бесперспективным

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

## Rule-based machine translation, RBMT

- Получает интенсивное развитие в 1970-х годах
- Словари + попытка посмотреть на то как работают лингвисты и вбить какие-то паттерны в компьютер (существительные оканчиваются на а-я и тп)
- Бывает разных видов

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

# Rule-based machine translation, RBMT

- Дословный перевод (Direct Machine Translation): делим текст по словам, переводим каждое, правим каждое слово в соответствии с накопленными правилами (окончания, падежи и тп). Правила придумывают лингвисты.
- Трансферные системы (Transfer-based Machine Translation): сначала выделяем синтаксические конструкции (сказуемые, подлежащие и тп), понимаем как слова надо переставить, а потом уже переводим.

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

## Эти типы стали есть на складе

- Придумывать правила вручную - сложно
- Огромное количество исключений
- Омонимия (разный смысл одних и тех же слов в зависимости от контекста)
- RBMT системы за годы холодной войны вышли на пик и успешно умерли, сегодня они не используются нигде

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

# Statistical machine translation (SMT)

- IBM, 1990 г.
- Берём корпус параллельных текстов, смотрим как часто слово house переводится как дом, строение, постройка, так и переводим!
- Работало лучше, чем всё что было до этого

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

# Statistical machine translation (SMT)

The screenshot shows a SMT interface with the following components:

- Top Bar:** Language selection tabs: ОПРЕДЕЛИТЬ ЯЗЫК, АНГЛИЙСКИЙ (selected), РУССКИЙ, НЕМЕЦКИЙ, РУССКИЙ (selected), АНГЛИЙСКИЙ, УКРАИНСКИЙ.
- Input Field:** The word "house" is entered in the English input field.
- Output Field:** The word "ЖИЛОЙ ДОМ" is displayed in the Russian output field, with a small checkmark icon.
- Intermediate Translations:** Below the main fields, the word "zhiloy dom" is shown.
- Bottom Buttons:** A microphone icon, a sound volume icon, a search bar placeholder "5/5000", and a set of icons for copy/paste, edit, and share.
- Left Panel (Definitions):**
  - house – определения**
  - Имя Существительное**
  - 1** a building for human habitation, especially one that is lived in by a family or small group of people.  
"Real foxes do, indeed, sometimes make their homes under human houses and, increasingly in this country at any rate, under city homes."
  - Синонимы:** residence, home, place of residence, homestead, a roof over one's head, habitation, dwelling (place), abode, domicile.
- Right Panel (Variants):**
  - house: варианты перевода**
  - Имя Существительное**
  - Частота** (Frequency) icon.
  - 1** дом, жилище, театр, здание, палата, гостиница, семья, рубка, хозяйство, род.
  - house, home, dwelling, door, premises, crib, housing, home, dwelling, house, abode, habitation, theater, house, playhouse, stage, living theater, theatre, building, house, structure, edifice, construction, fabric, chamber, ward, house, hotel, inn, hostel, guesthouse, pub, house, family, household, home, kin, colony, house, felling, cutting, deckhouse, house, chopping, economy, farm, household, property, house, establishment, genus, race, kind, family, type, house.**

## Word-based SMT

- Начали конечно же со статистического перевода по отдельным словам
- Пример реализации на python: <https://github.com/shawa/IBM-Model-1>
- Дальше попробовали также по статистике переставлять слова, добавлять недостающие артикли
- Всё ещё много проблем с омонимией и согласованностью слов в предложениях

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

# GTA San Andreas (2005)



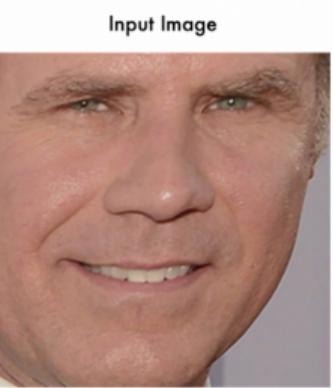
## Phrase-based SMT

- Word-based SMT был основан на мешке слов, тут подмешали N-граммы
- С 2006 года этот подход использовали абсолютно все
- Так продолжалось до 2016 года
- В 2016 году Google перевернул игру:  
<https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>

[https://vas3k.ru/blog/machine\\_translation/](https://vas3k.ru/blog/machine_translation/)

# Нейросетевой перевод

# Энкодер для картинки



128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.1595809660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.08485303741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.00727775558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.1340749858099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788132833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

# Энкодер для текста

Input Sentence

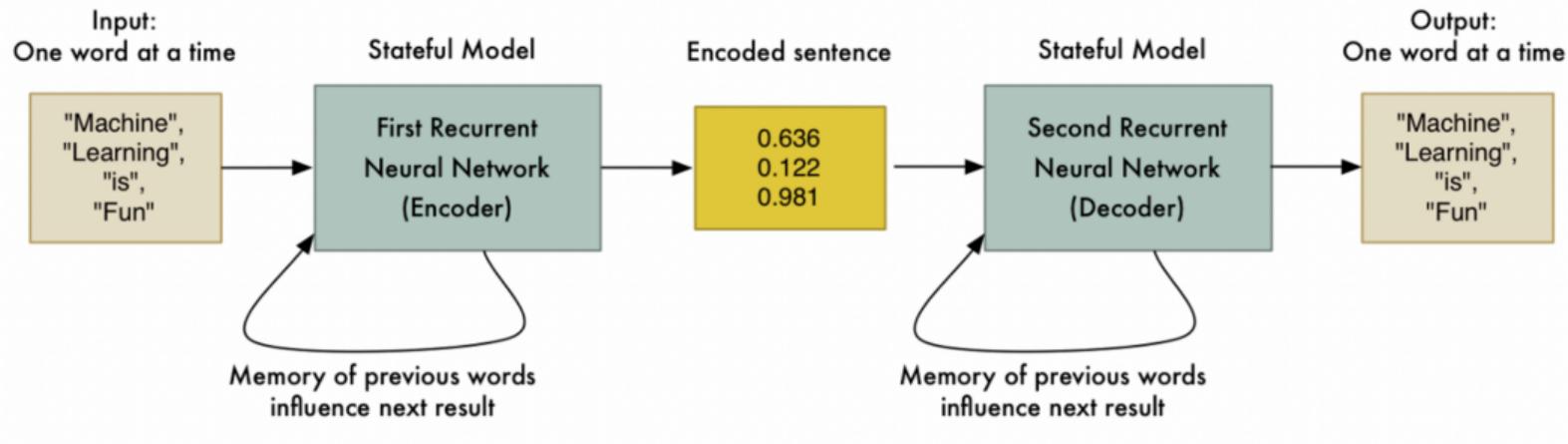
"Machine Learning is Fun!"



Measurements Generated from Sentence

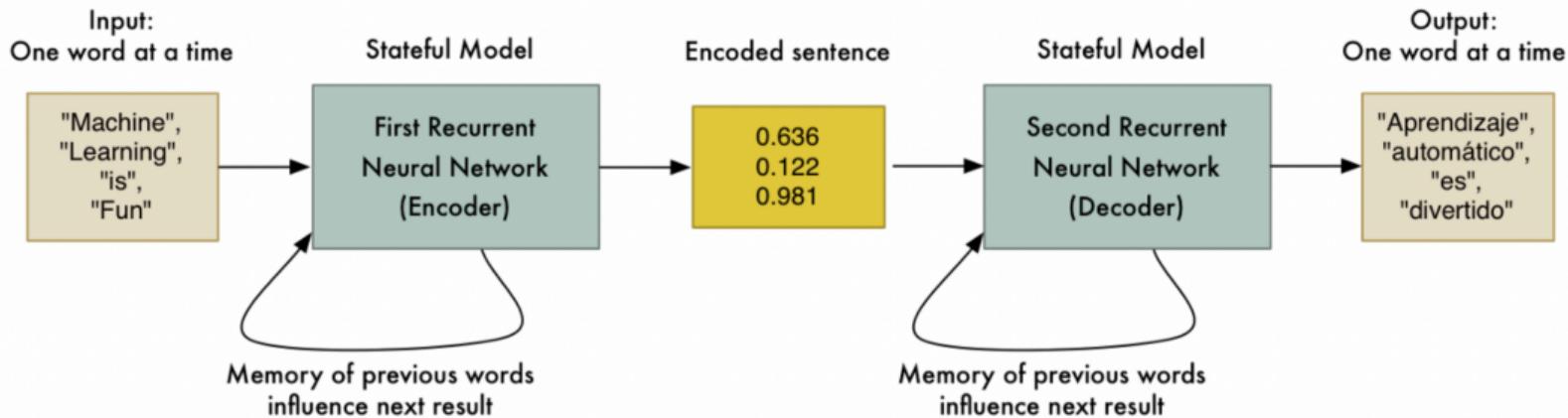
0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.0974868883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.2105591947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.0723776452386379
-0.12216668576002	-0.007277755558491	-0.0369012914579799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154737377	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11090479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

# Автокодировщик для текста



[https://medium.com/@ageitgey/  
build-your-own-google-translate-quality-machine-translation-system-d7dc274bd476](https://medium.com/@ageitgey/build-your-own-google-translate-quality-machine-translation-system-d7dc274bd476)

# Автокодировщик для текста



[https://medium.com/@ageitgey/  
build-your-own-google-translate-quality-machine-translation-system-d7dc274bd476](https://medium.com/@ageitgey/build-your-own-google-translate-quality-machine-translation-system-d7dc274bd476)

# Нейросетевые переводы

- Энкодер пытается понять смысл и закодировать текст
- Декодер пытается раскодировать текст и сгенерировать таргет

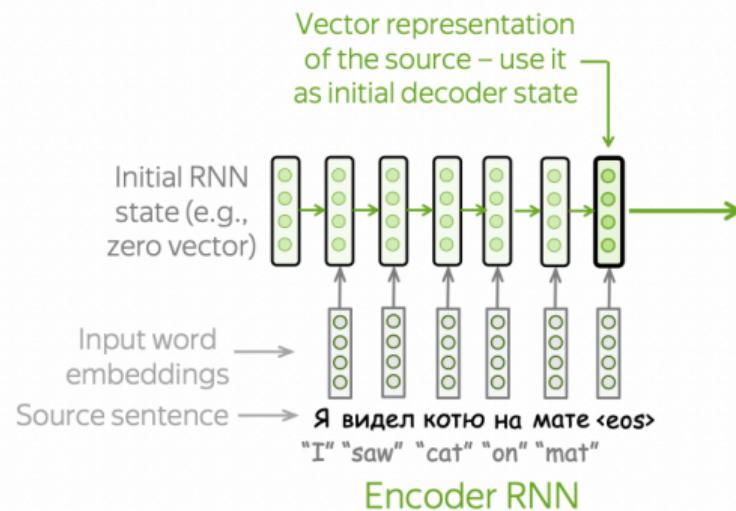
# Нейросетевые переводы

- Архитектуры переводчиков очень разные
- Поначалу это были RNN, после перешли на многослойные двунаправленные LSTM
- Дальше стали пробовать разные виды эмбеддингов и механизмы внимания
- Сегодня автопереводы делаются преимущественно трансформерами

## Нейросети + catboost

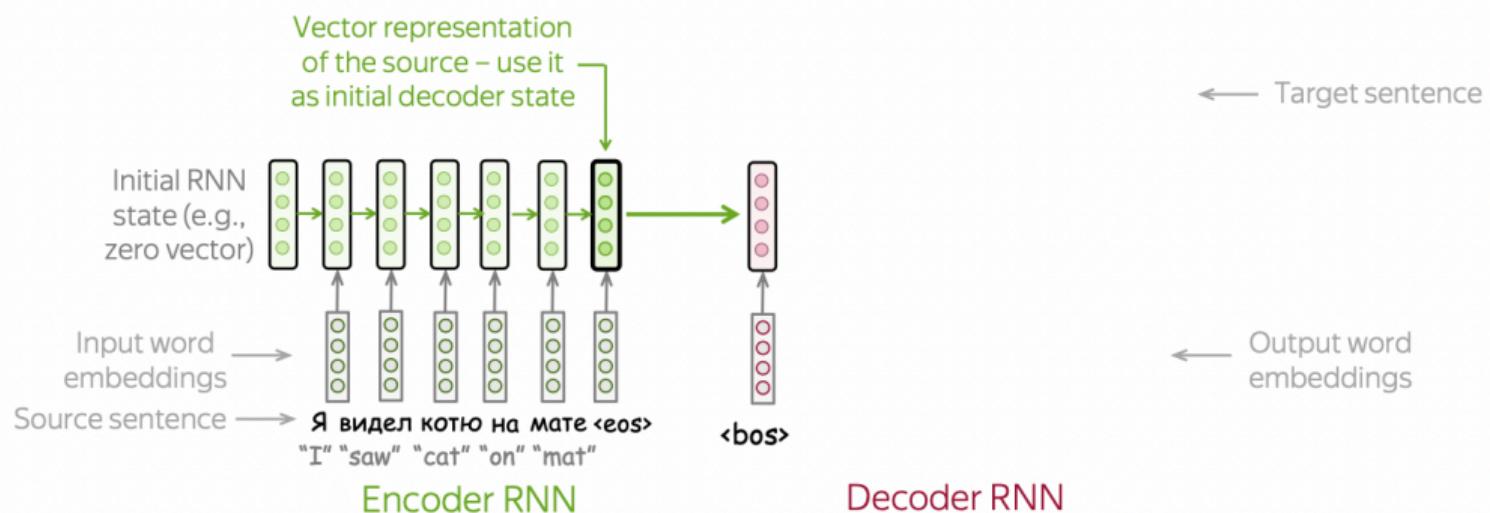
- На коротких фразах нейросетевой перевод работает плохо
- Перевод идёт двумя способами: статистическим и нейросетевым, а после Catboost выбирает тот, который лучше подходит
- Пресс-релиз Яндекса от 2017: <https://yandex.ru/blog/company/kak-pobedit-mornikov-yandeks-zapustil-gibridnuyu-sistemu-perevoda>
- Статья Яндекса на хабре про историю автоперевода:  
<https://habr.com/ru/company/yandex/blog/224445/>

# Простейшая модель: RNN энкодер-декодер



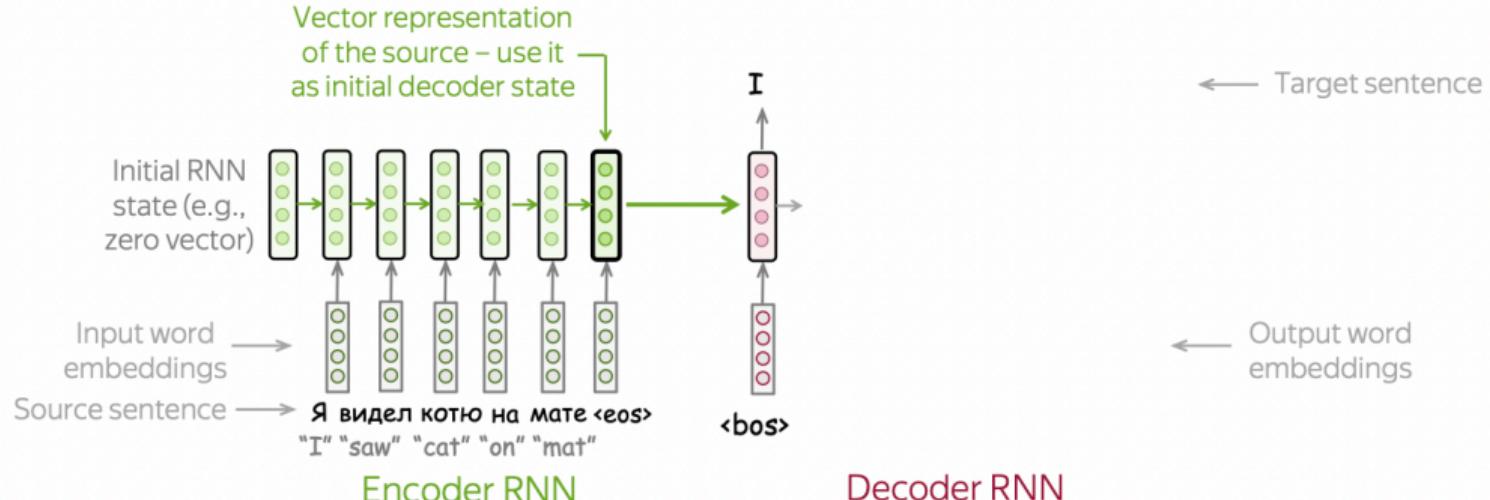
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Простейшая модель: RNN энкодер-декодер



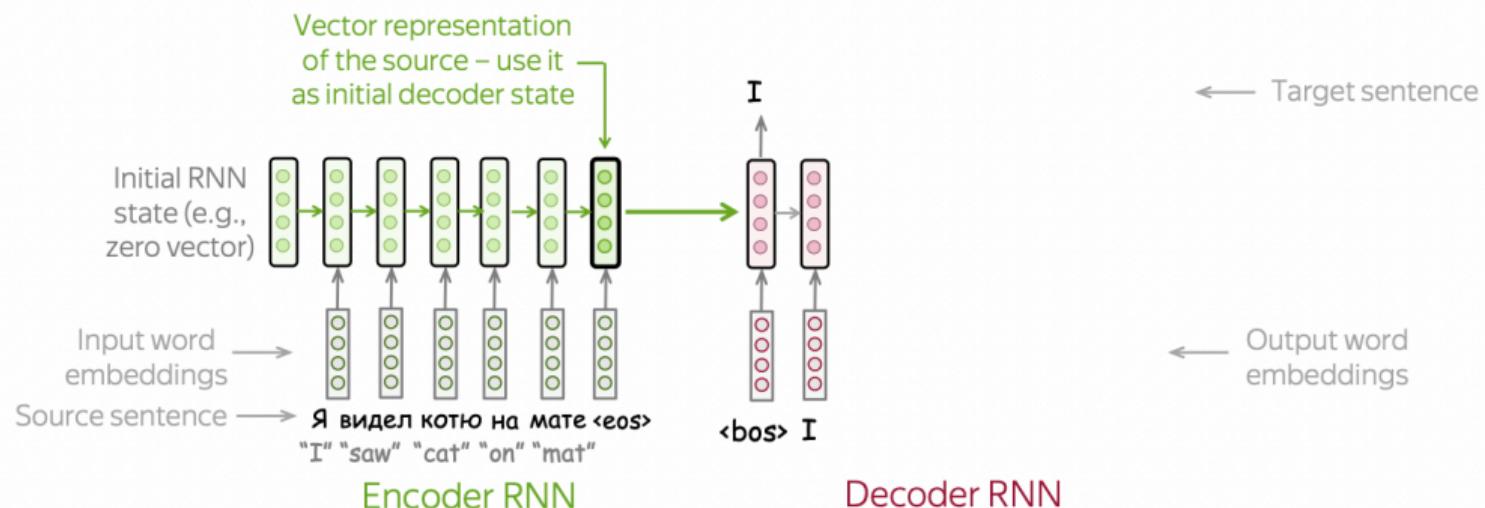
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Простейшая модель: RNN энкодер-декодер



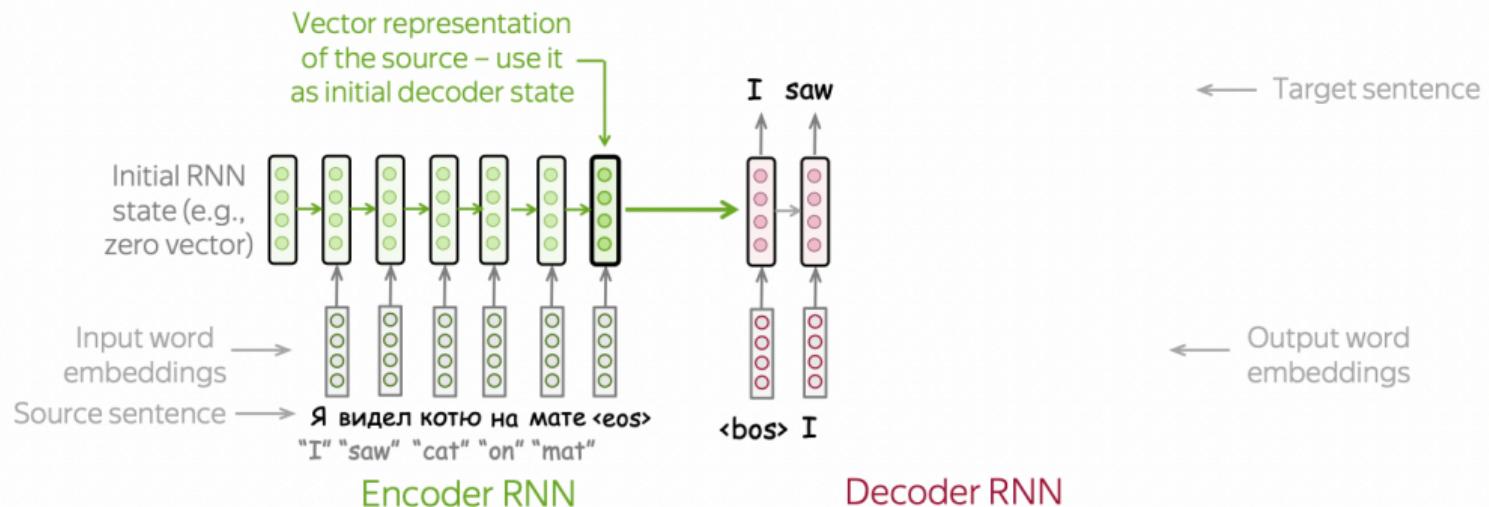
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Простейшая модель: RNN энкодер-декодер



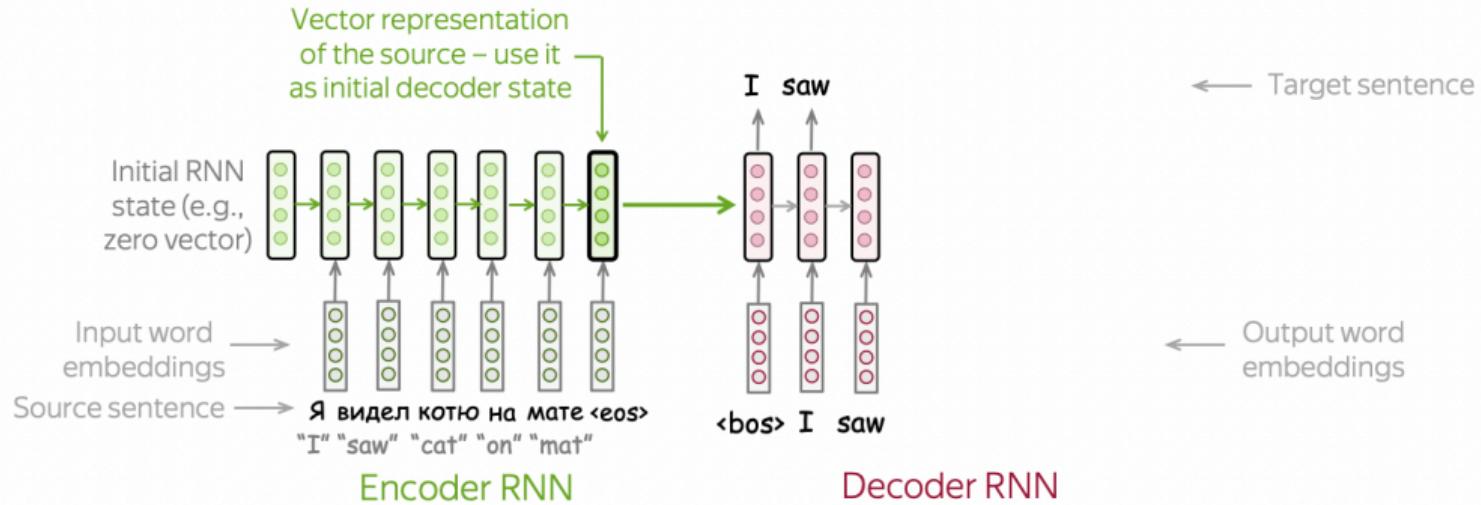
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Простейшая модель: RNN энкодер-декодер



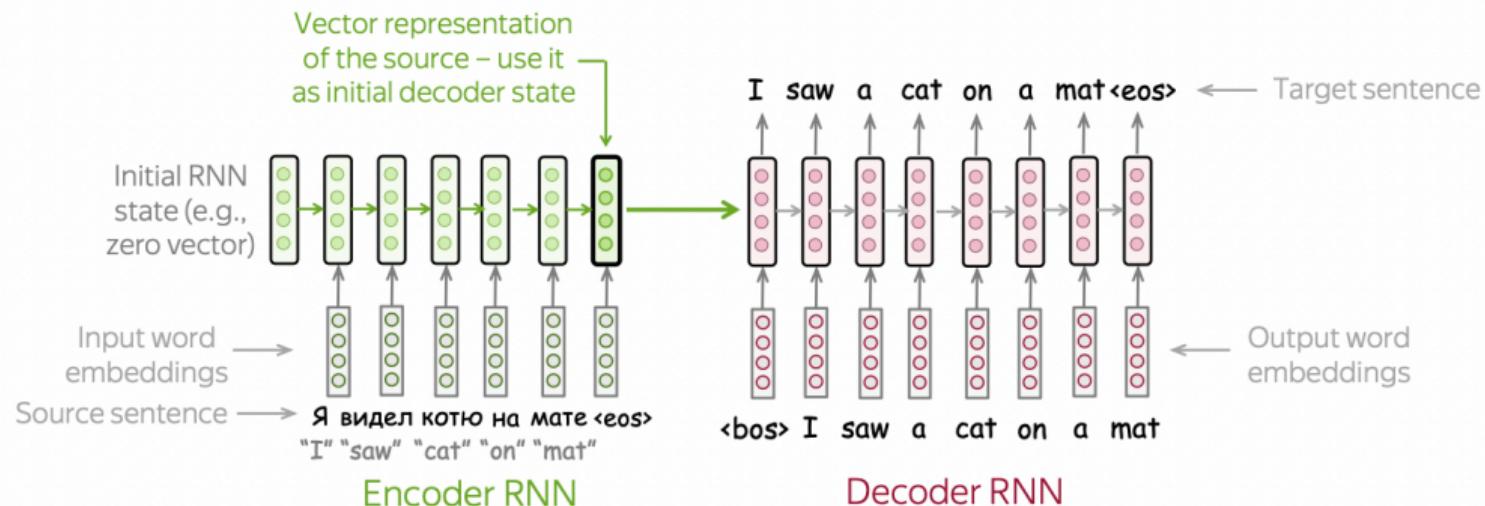
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Простейшая модель: RNN энкодер-декодер



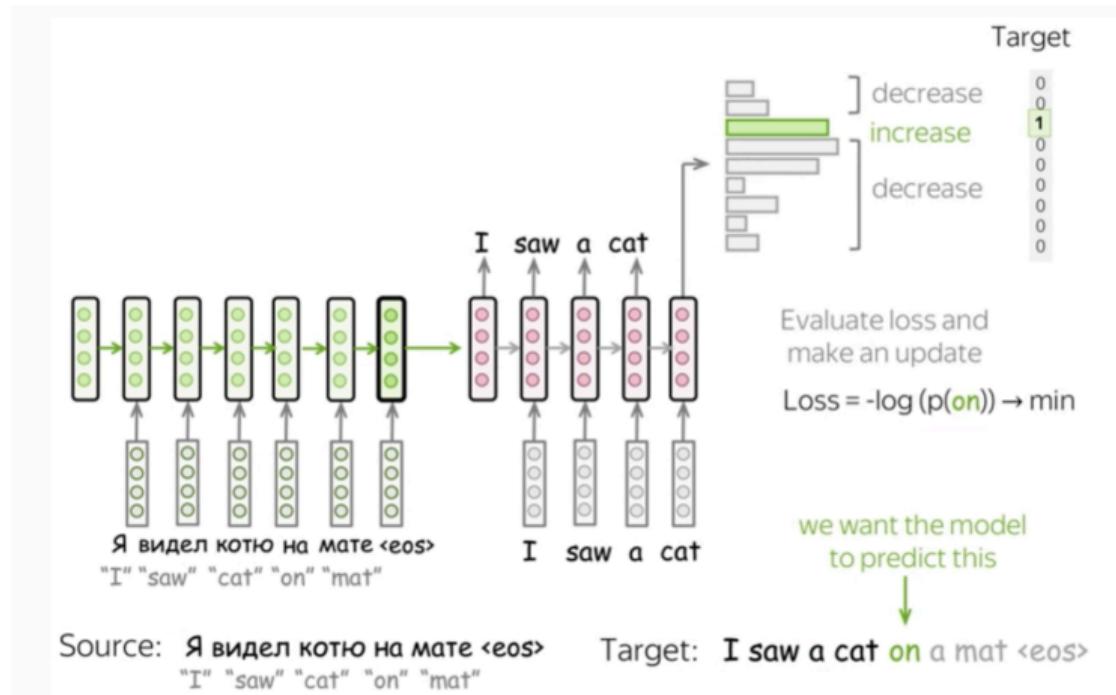
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Простейшая модель: RNN энкодер-декодер



[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Как обучить модель?



[https://lena-voita.github.io/resources/lectures/seq2seq/general/seq2seq\\_training\\_with\\_target.mp4](https://lena-voita.github.io/resources/lectures/seq2seq/general/seq2seq_training_with_target.mp4)

# Как сделать прогноз?

- Жадно (greedy decoding): на каждом шаге берём токен с самой большой вероятностью

$$\prod_{t=1}^T \arg \max_{y_t} p(y_t | y_{<t}, x) \neq \arg \max_y \prod_{t=1}^T p(y_t | y_{<t}, x)$$

- Наш прогноз это последовательность, жадный способ на каждом шаге выбирает локальный оптимум
- Перебирать все траектории, чтобы найти глобальный оптимум очень дорого

# Неоптимальность Greedy Decoding

Time step	1	2	3	4
A	0.5	0.1	0.2	0.0
B	0.2	0.4	0.2	0.2
C	0.2	0.3	0.4	0.2
<eos>	0.1	0.2	0.2	0.6

$$P = 0.5 \times 0.4 \times 0.4 \times 0.6 = 0.048$$

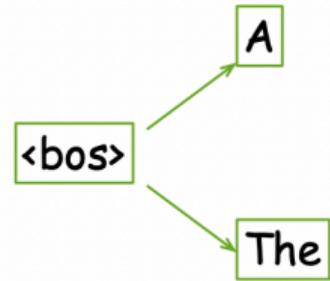
Time step	1	2	3	4
A	0.5	0.1	0.1	0.1
B	0.2	0.4	0.6	0.2
C	0.2	0.3	0.2	0.1
<eos>	0.1	0.2	0.1	0.6

$$P = 0.5 \times 0.3 \times 0.6 \times 0.6 = 0.054$$

# Beam Search

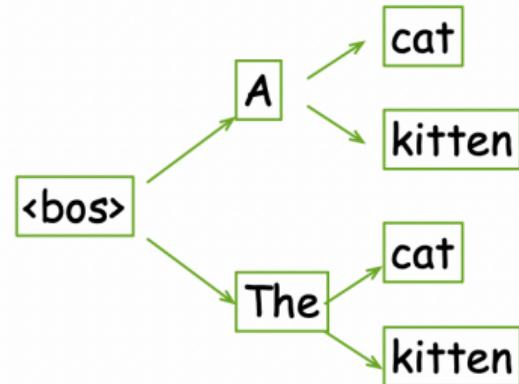
- Давайте поддерживать несколько самых вероятных траекторий
- Такая стратегия называется **Beam Search**
- Число траекторий, которое мы помним будет гиперпараметром (больше 10 брать не стоит)

# Beam Search



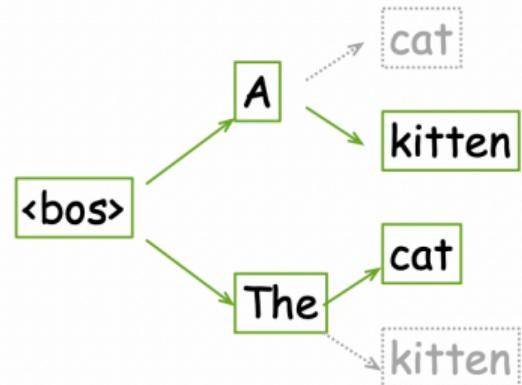
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



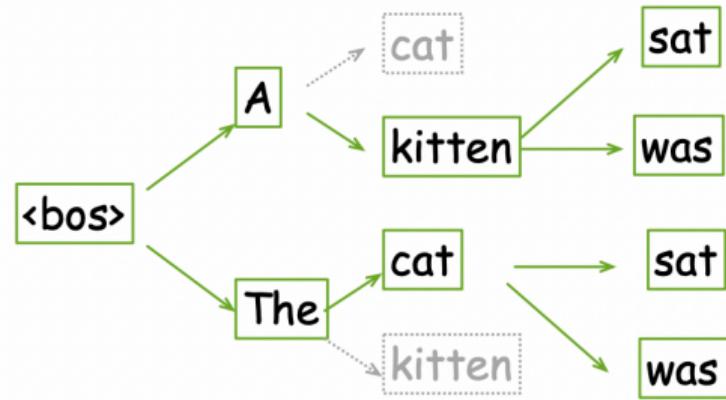
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



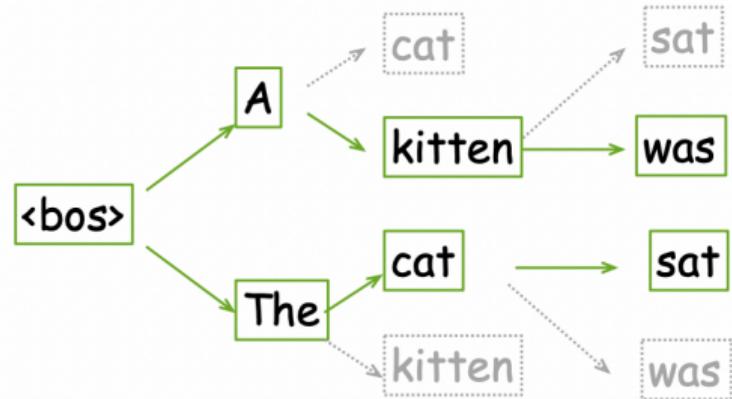
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



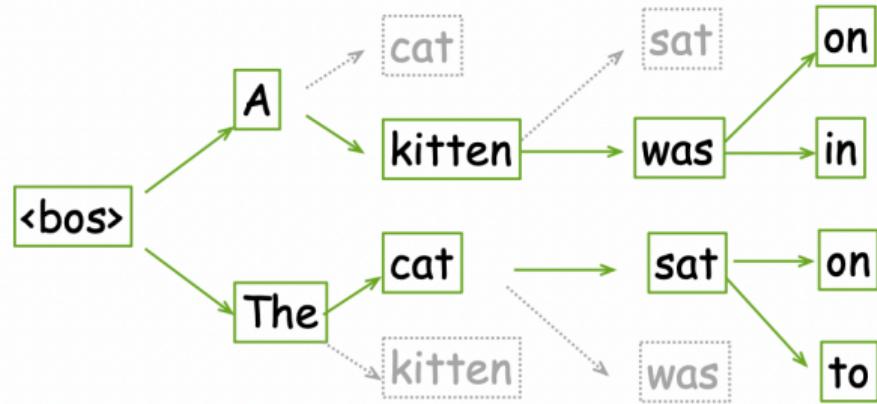
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



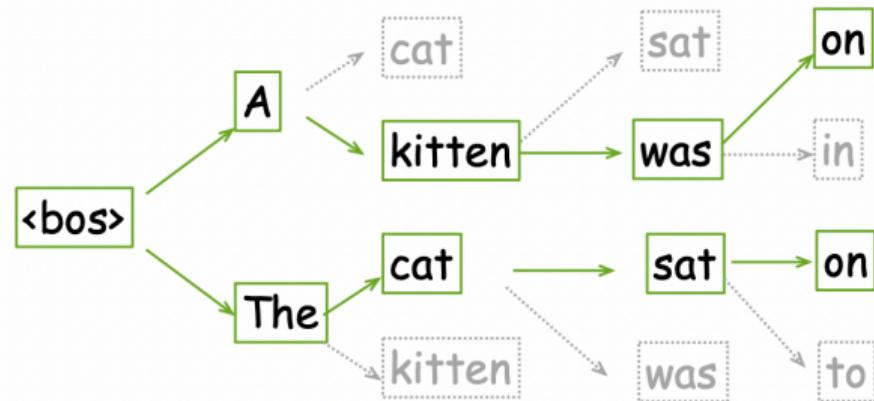
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

## Beam Search



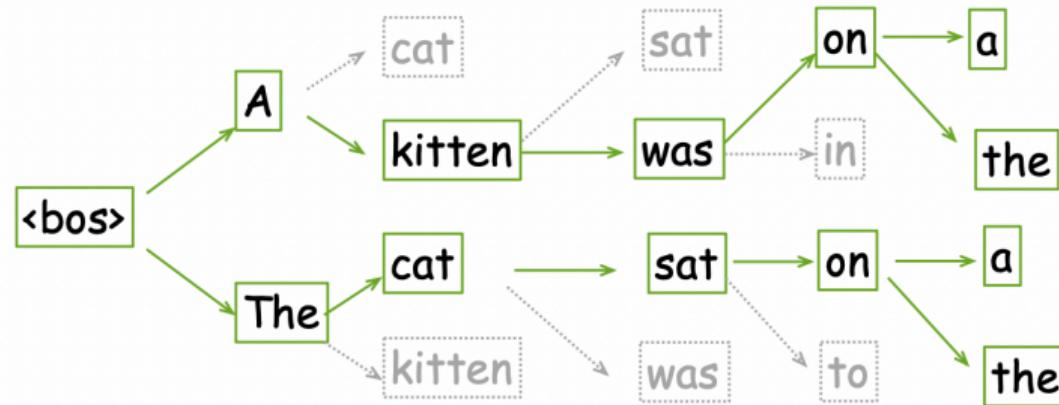
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



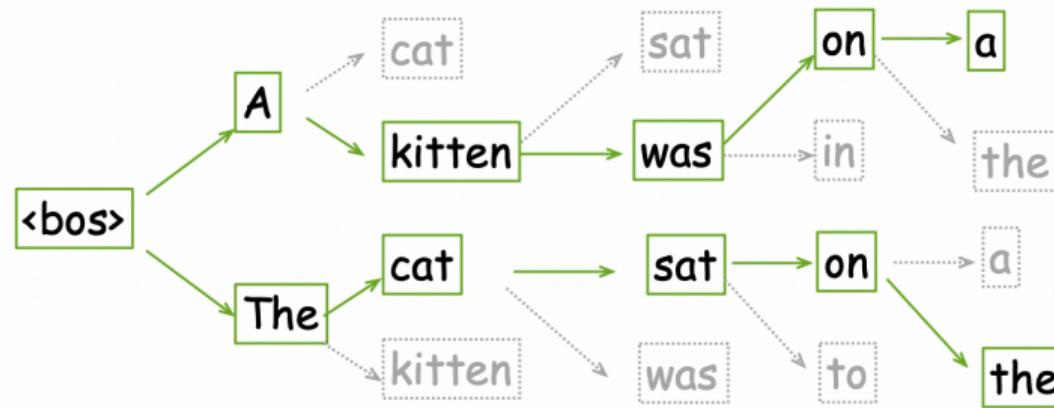
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



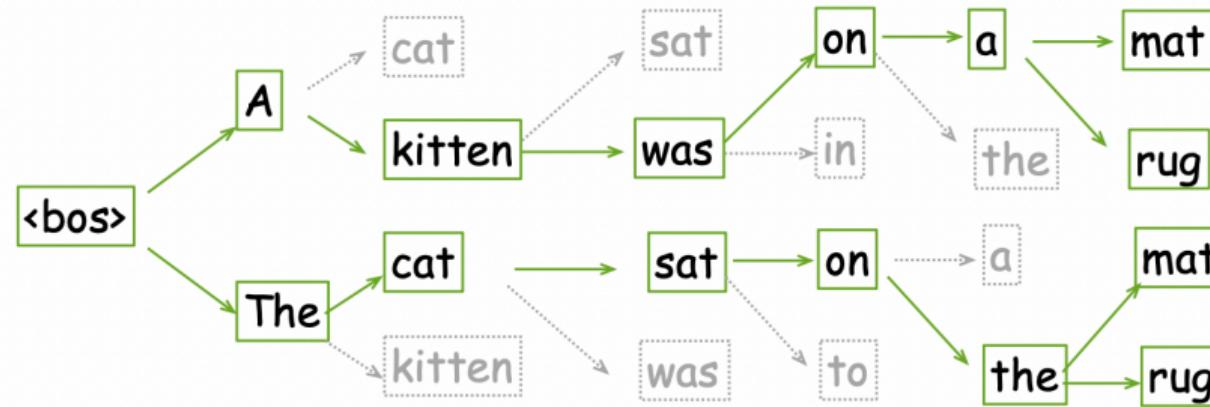
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



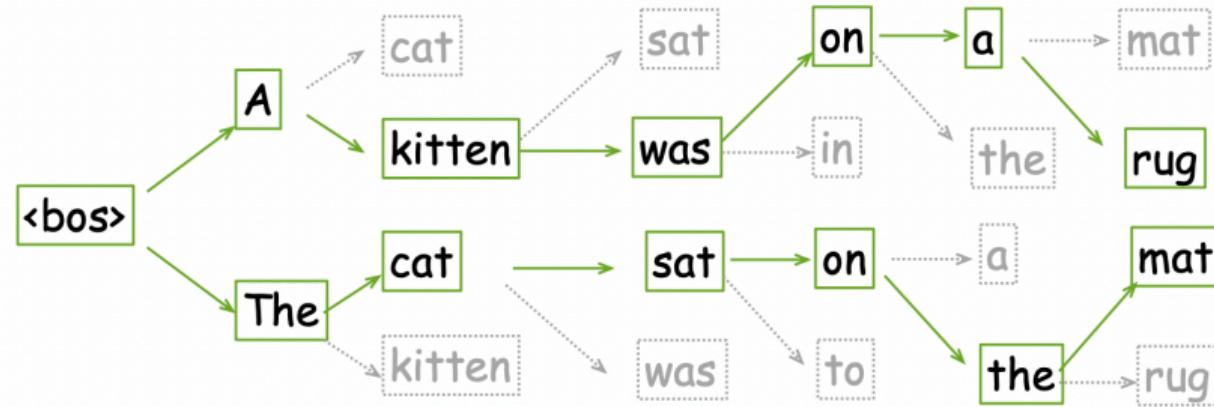
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



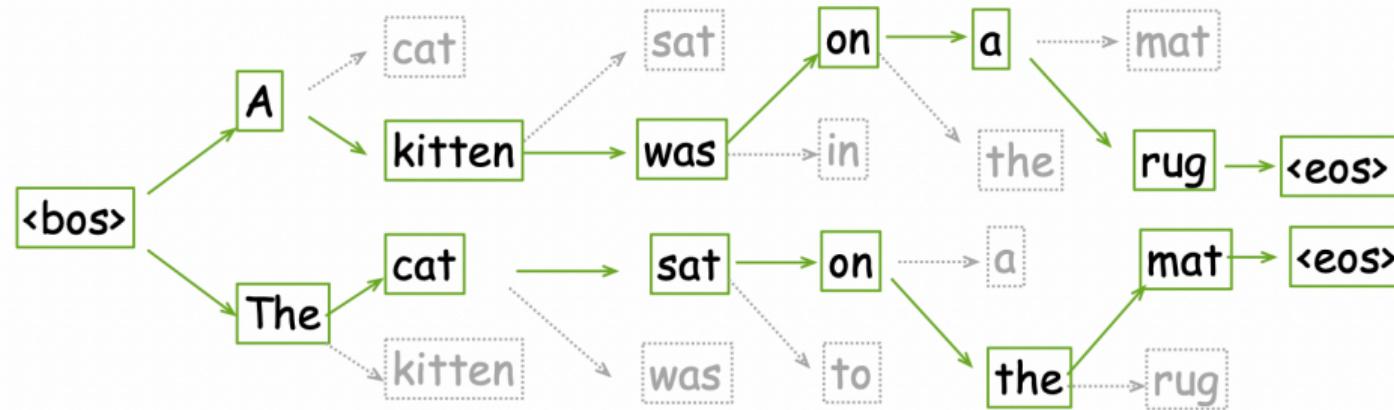
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



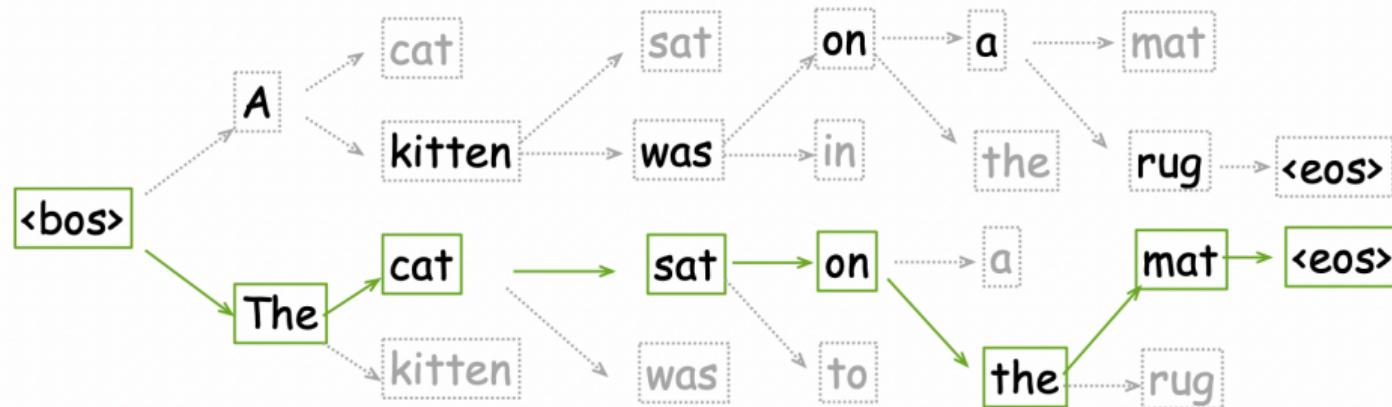
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Beam Search



[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

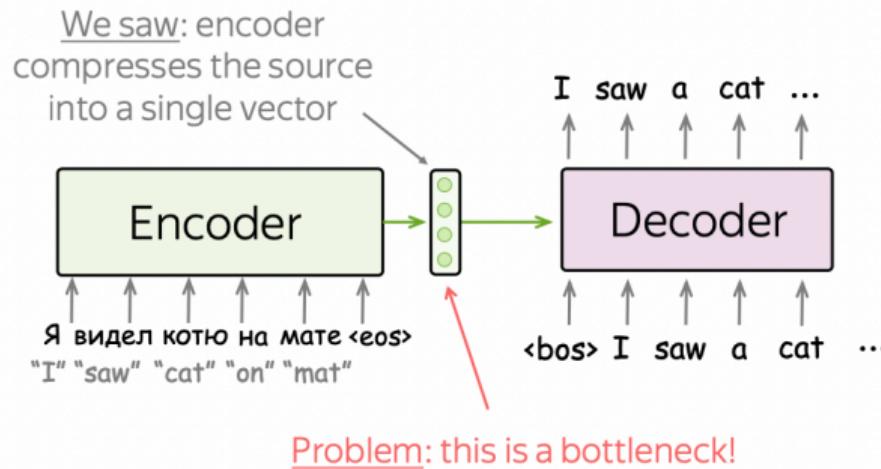
# Beam Search



[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Механизмы внимания

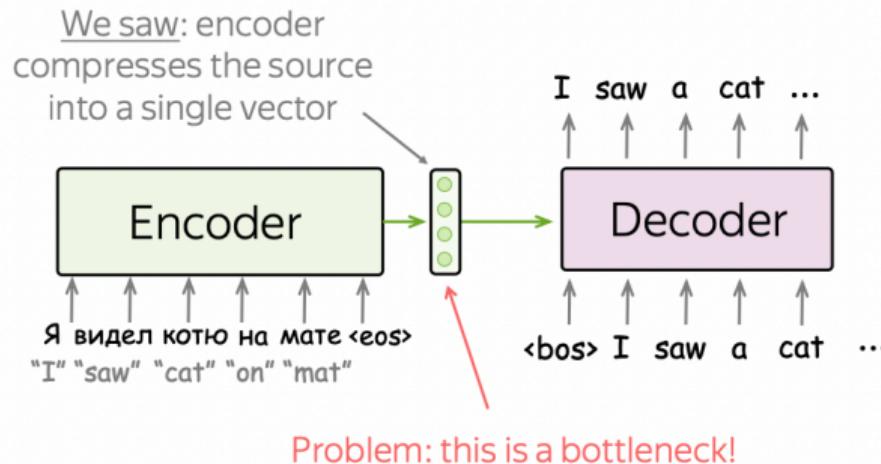
# В чём проблема простой RNN модели?



- Сжать предложение в вектор сложно
- Декодеру может понадобиться информация о разных частях предложения для его расшифровки

[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

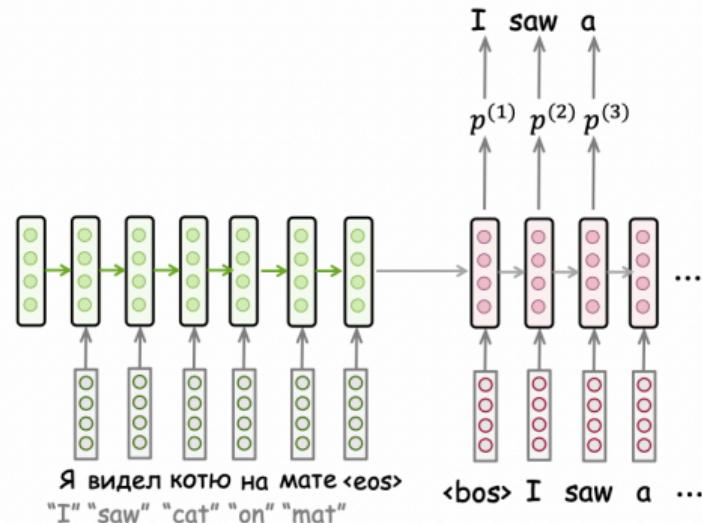
# В чём проблема простой RNN модели?



- **Attention:** на разных шагах надо позволить модели фокусироваться на разных входных токенах

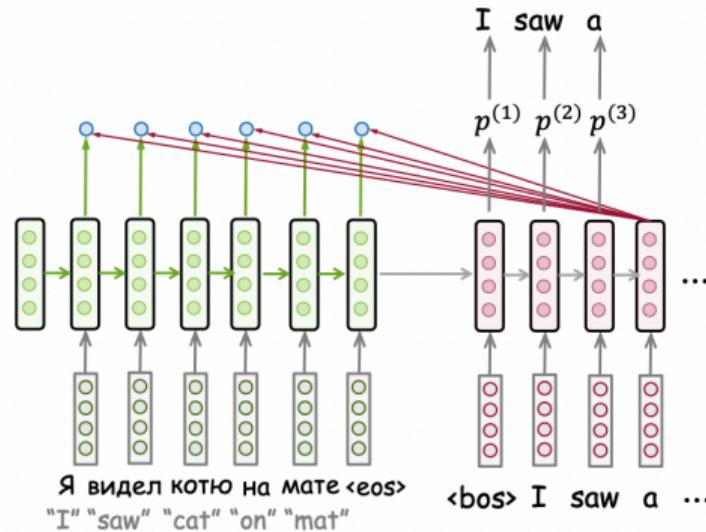
[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Attention

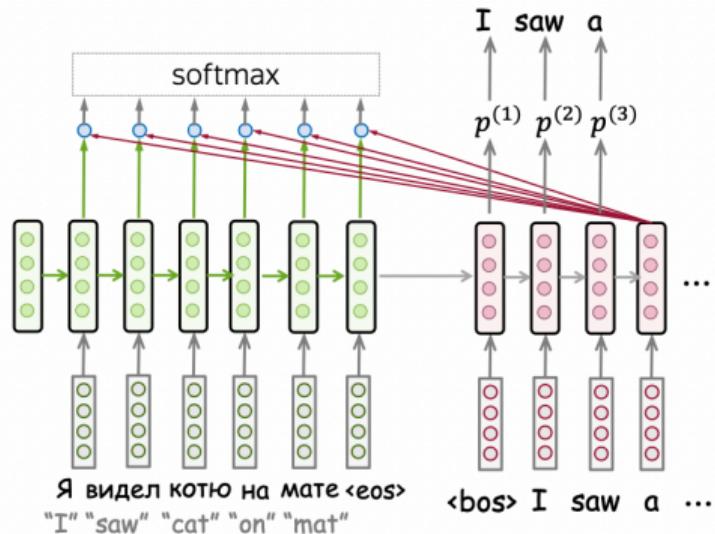


[https://github.com/yandexdataschool/nlp\\_course/tree/2021/week04\\_seq2seq](https://github.com/yandexdataschool/nlp_course/tree/2021/week04_seq2seq)

# Attention

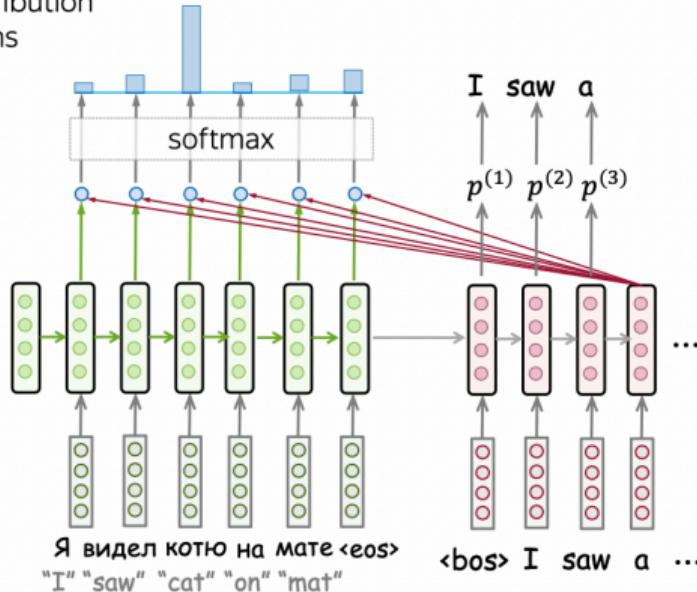


# Attention



# Attention

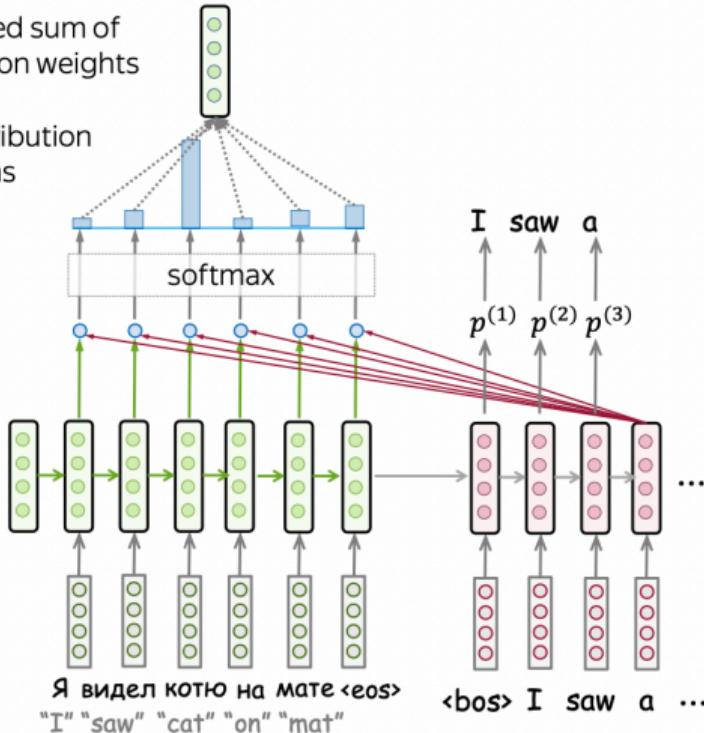
Attention weights: distribution over source tokens



# Attention

Attention output: weighted sum of encoder states with attention weights

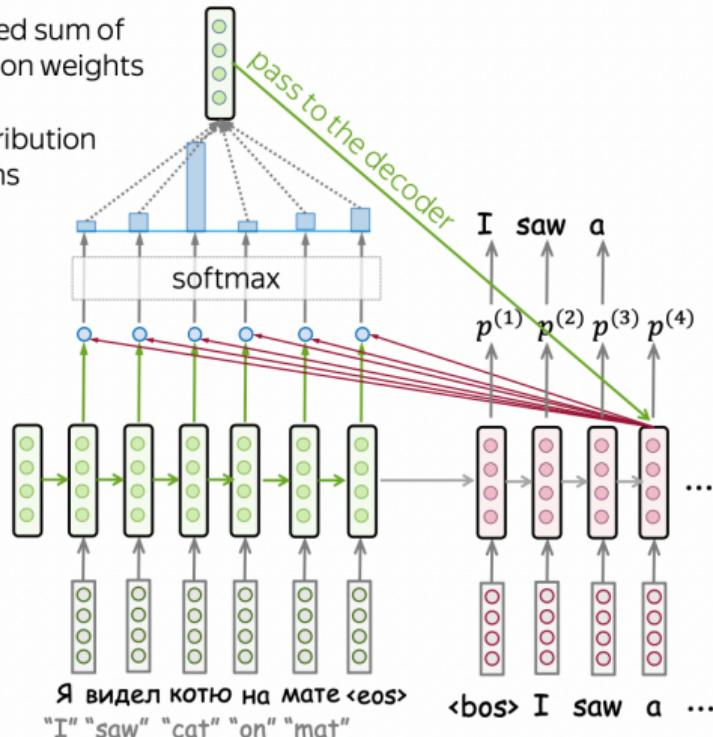
Attention weights: distribution over source tokens



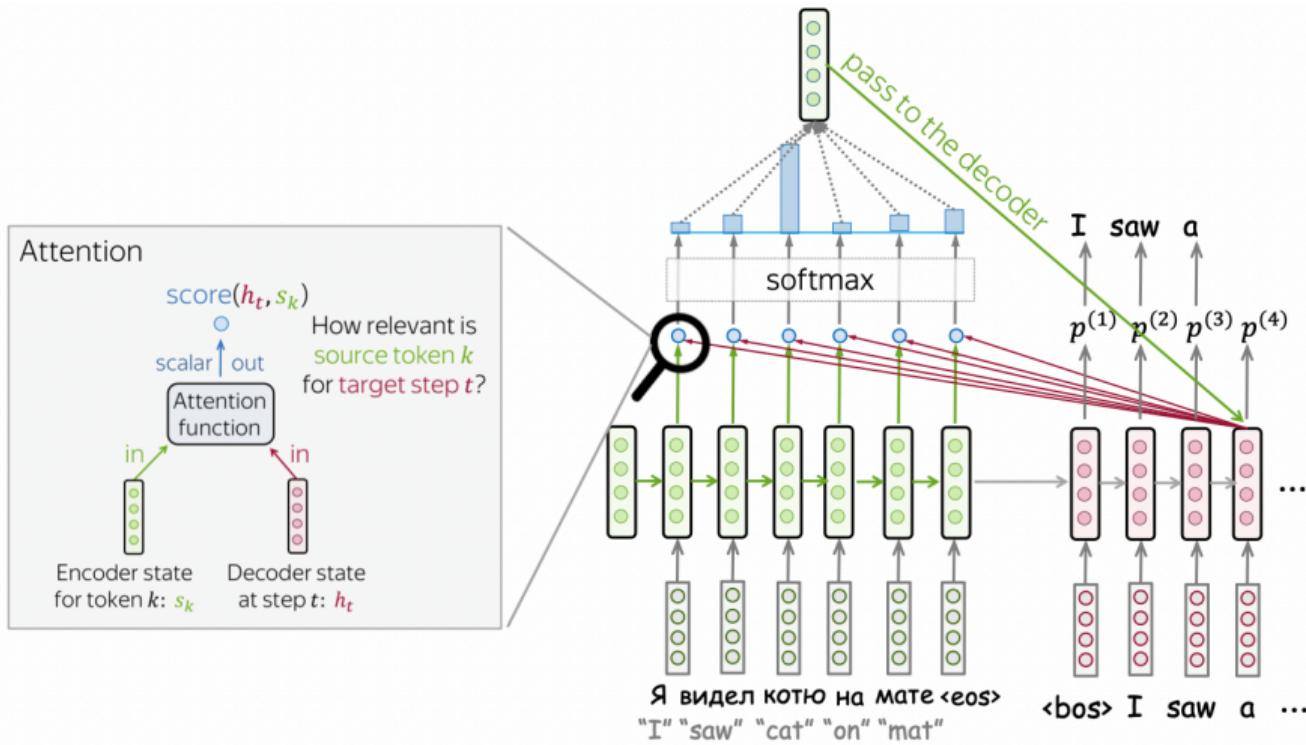
# Attention

Attention output: weighted sum of encoder states with attention weights

Attention weights: distribution over source tokens



# Attention



# Computation Pipeline

Attention input

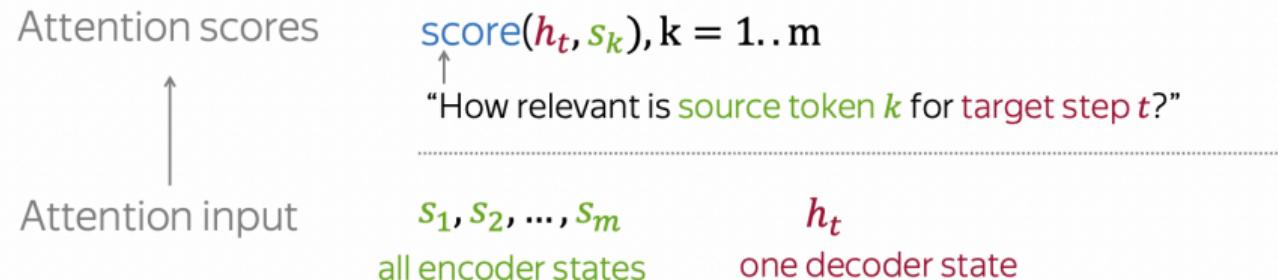
$s_1, s_2, \dots, s_m$

all encoder states

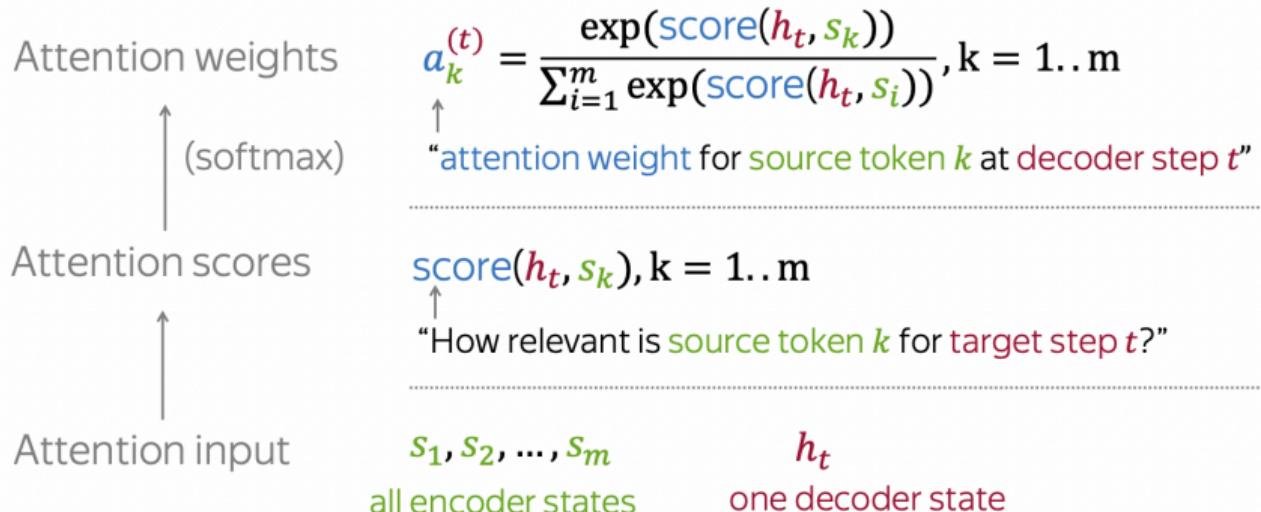
$h_t$

one decoder state

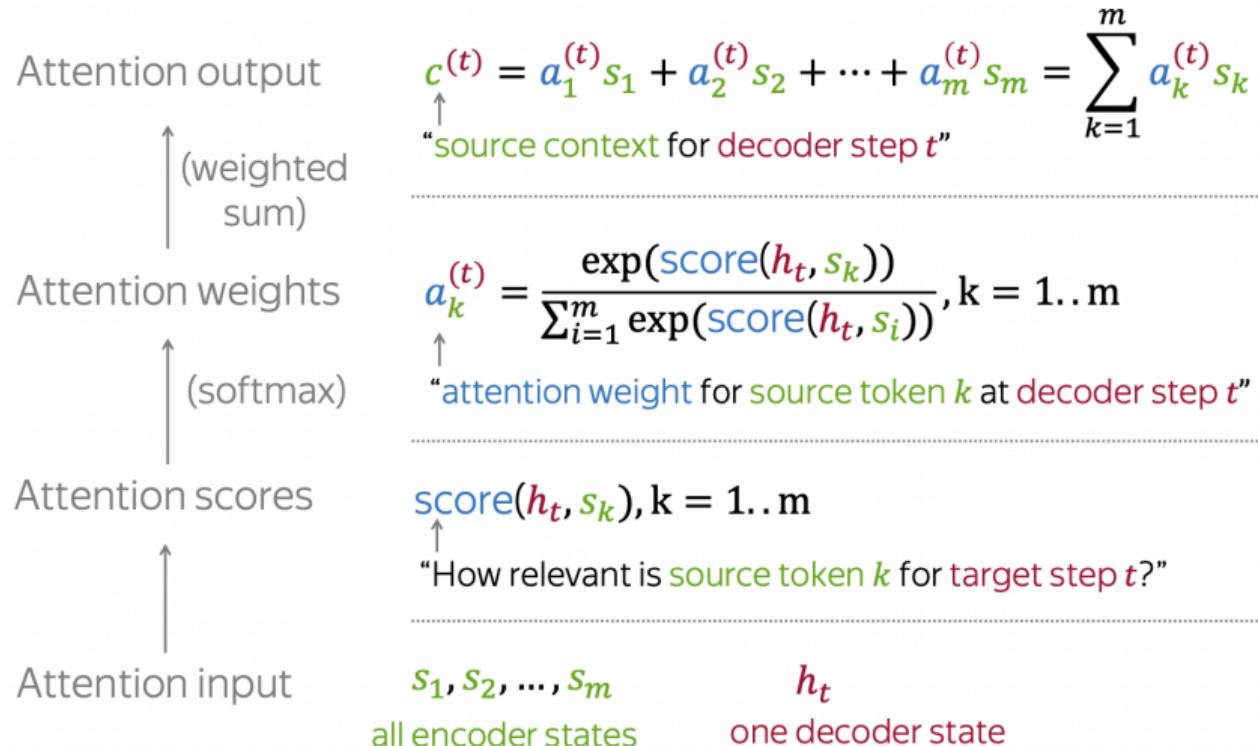
# Computation Pipeline



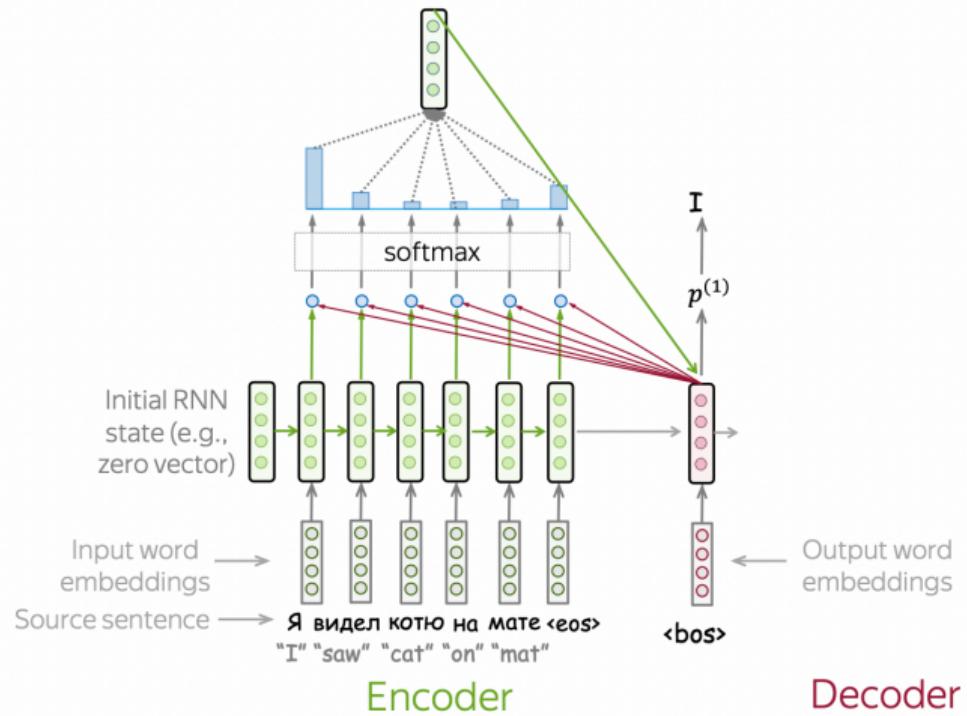
# Computation Pipeline



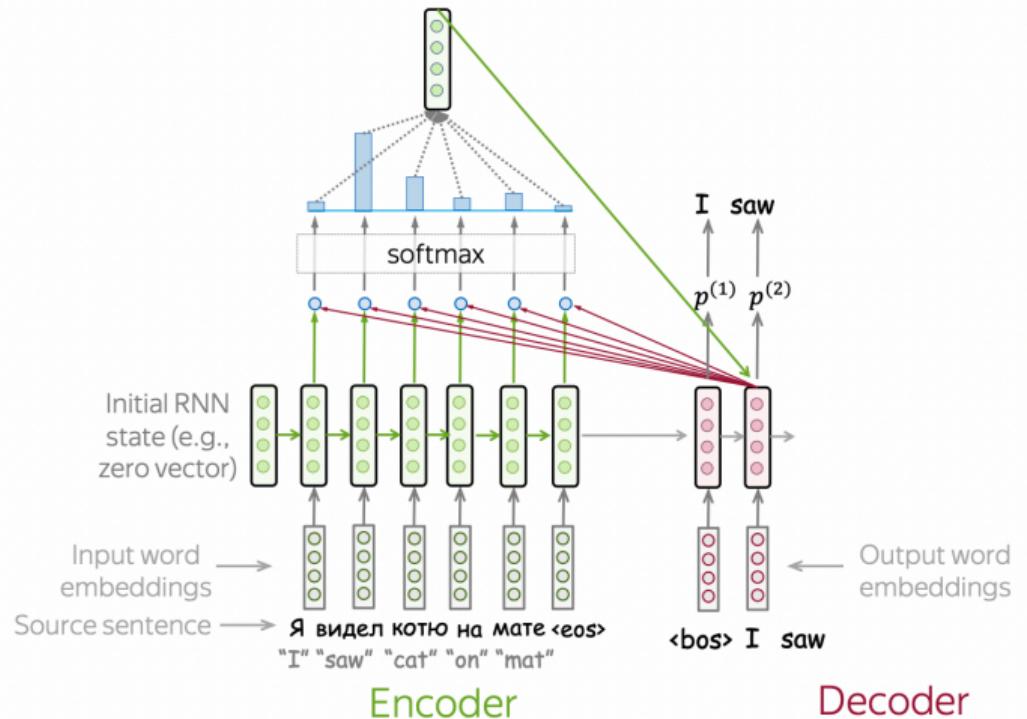
# Computation Pipeline



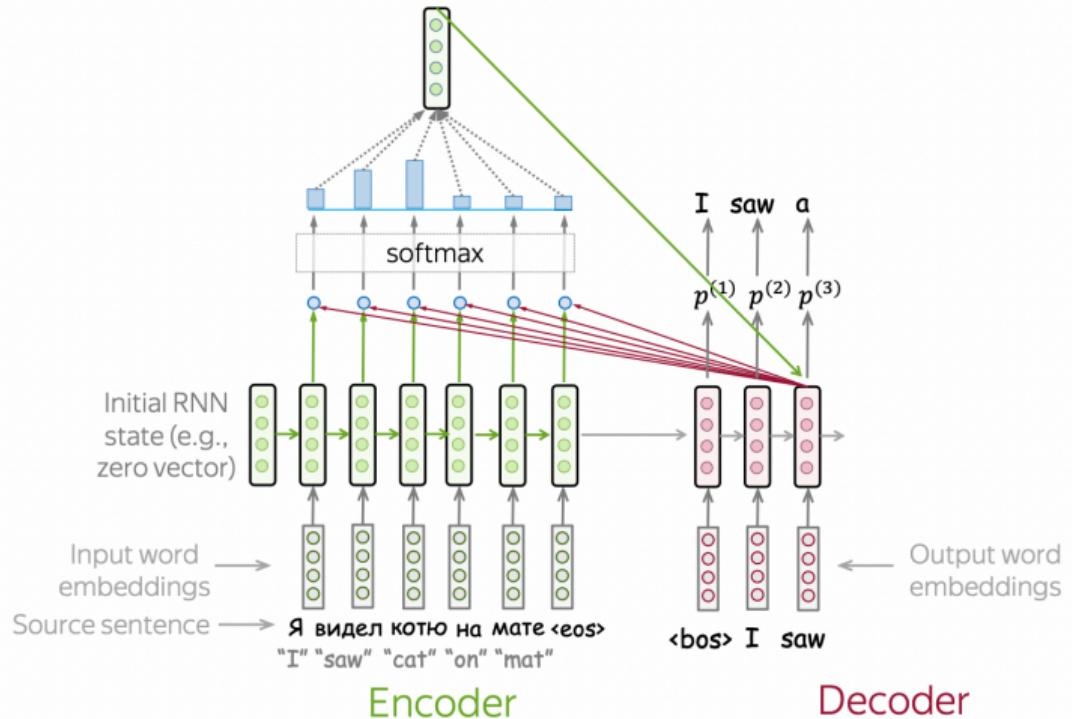
# Внимание в декодере



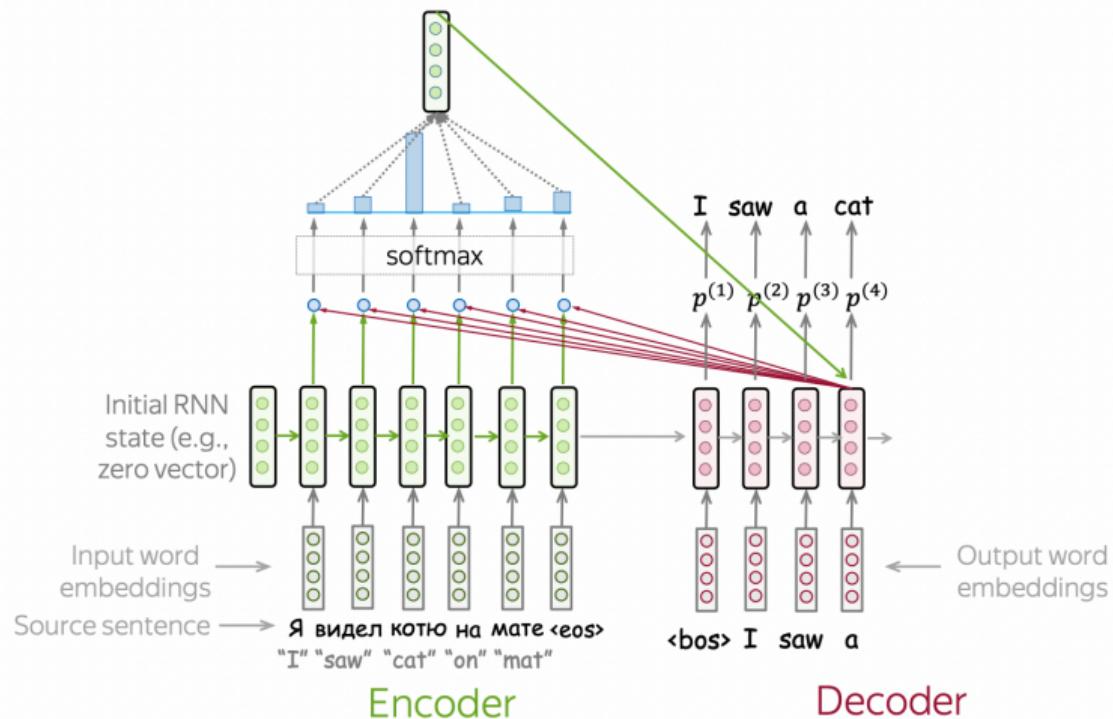
# Внимание в декодере



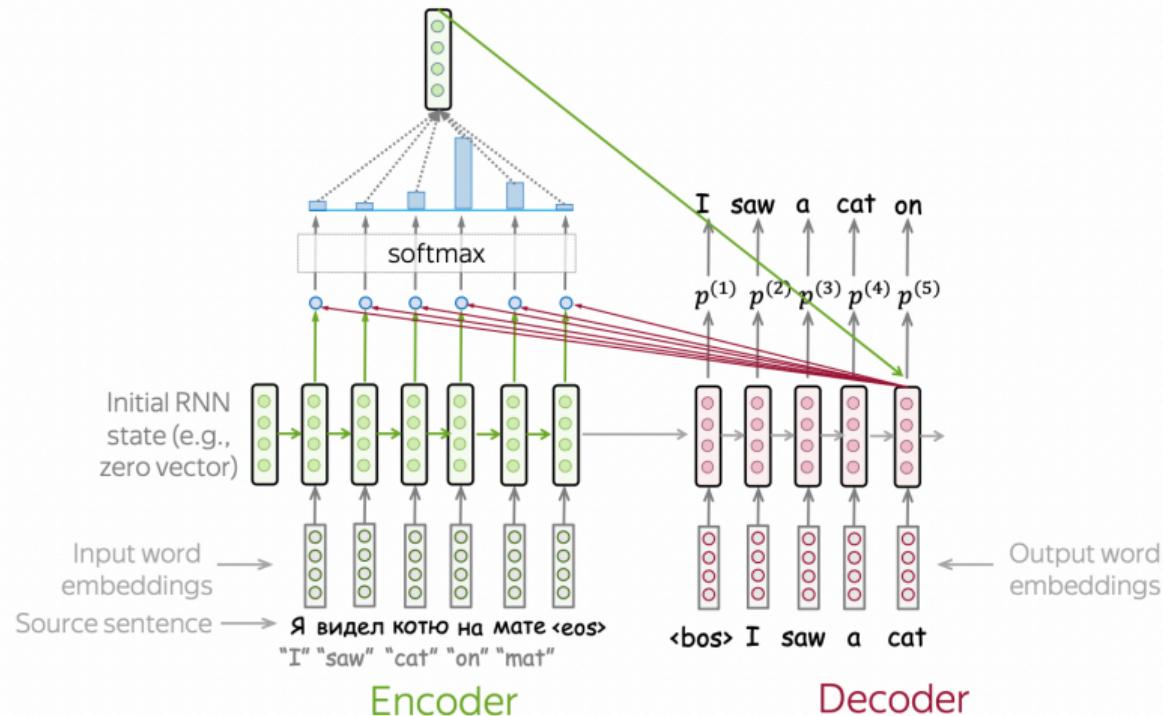
# Внимание в декодере



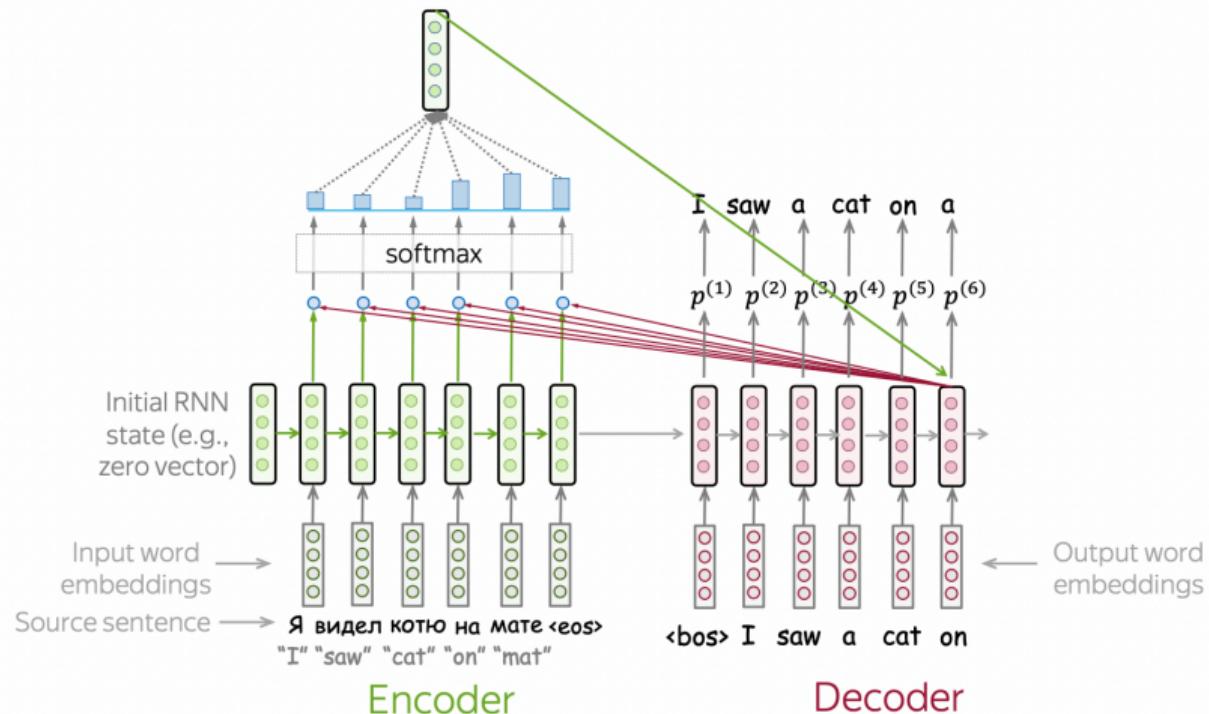
# Внимание в декодере



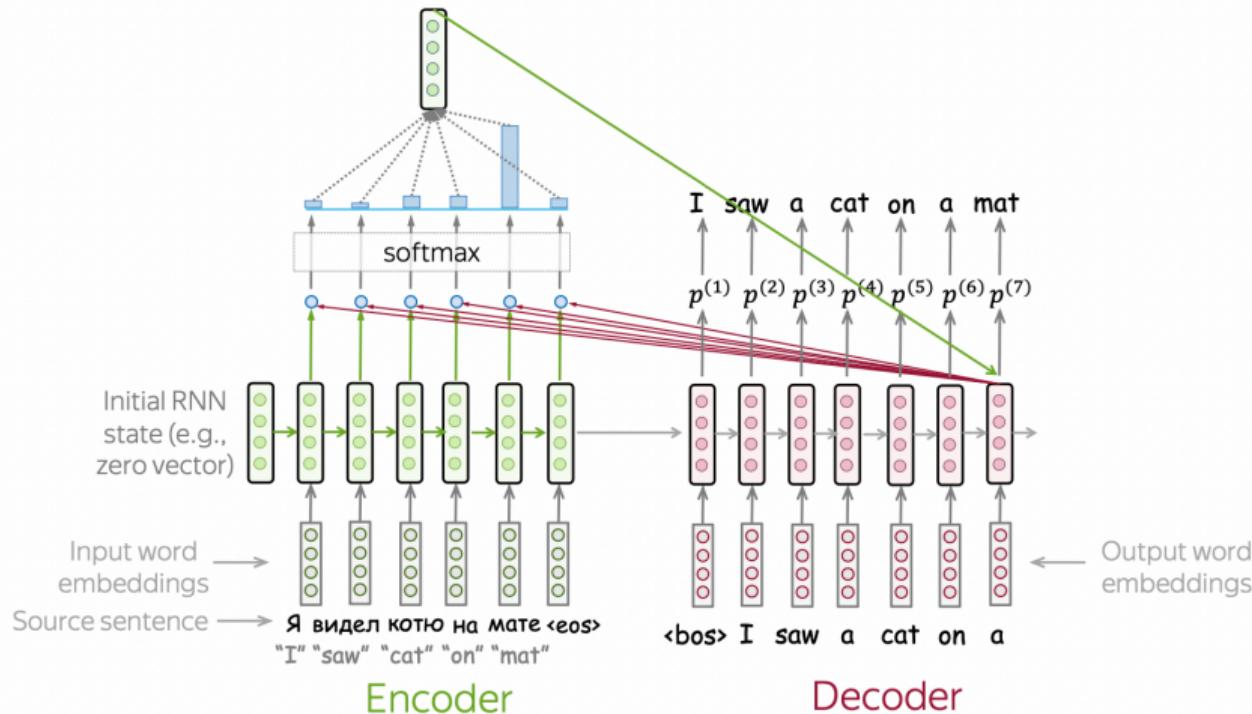
# Внимание в декодере



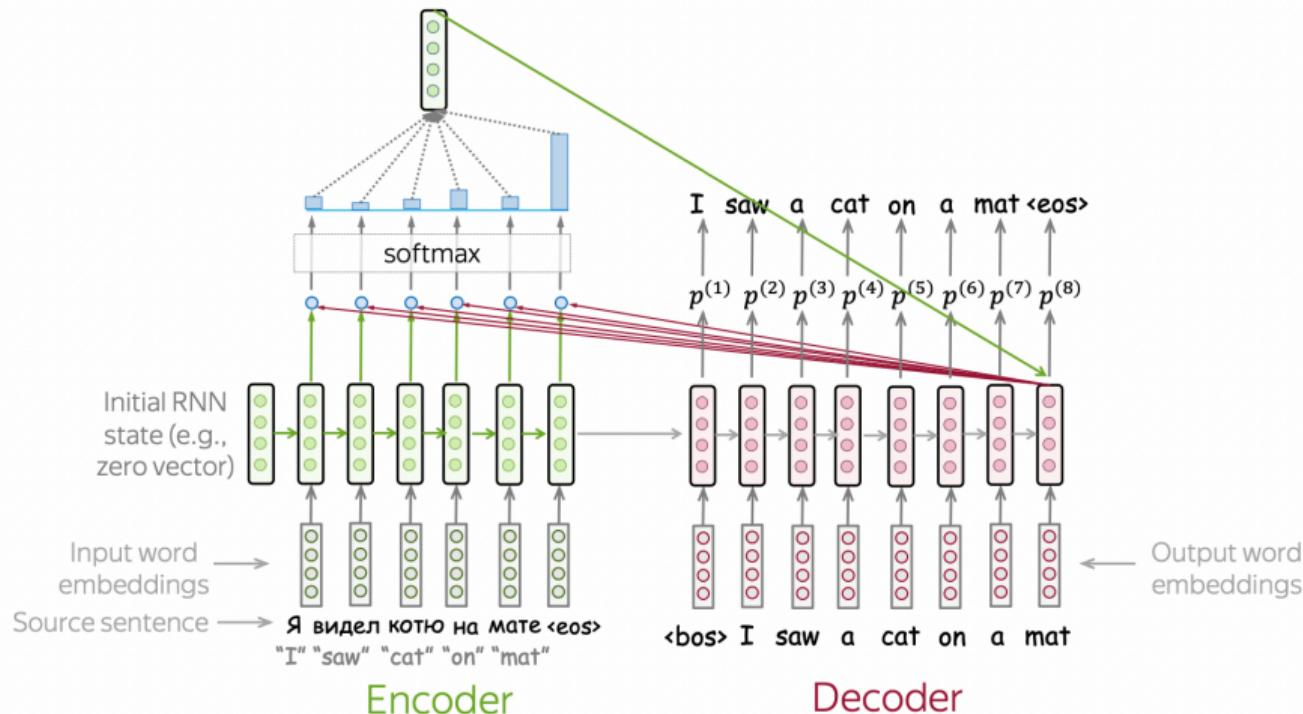
# Внимание в декодере



# Внимание в декодере



# Внимание в декодере



# Attention Score Functions

## Attention

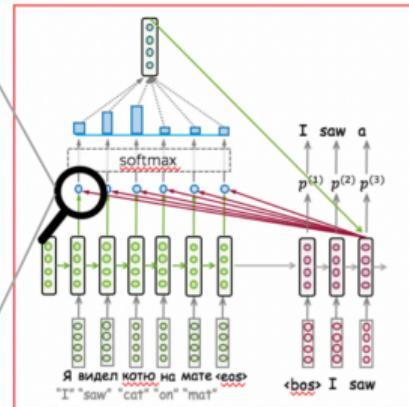
$\text{score}(h_t, s_k)$

scalar  
out

Attention  
function



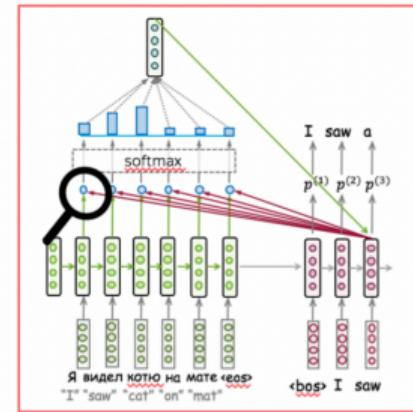
How relevant is  
source token  $k$   
for target step  $t$ ?



# Attention Score Functions

- Dot-product:  $\text{score}(h_t, s_k) = h_t^T s_k$

$$\begin{matrix} h_t^T \\ \text{---} \end{matrix} \times \begin{matrix} s_k \\ \text{---} \end{matrix}$$



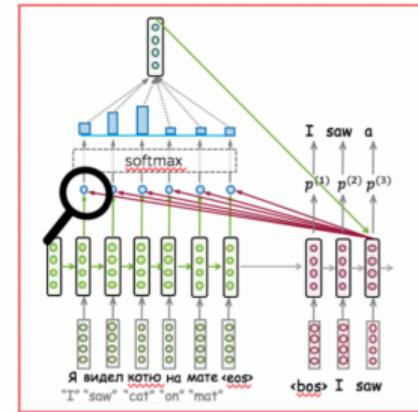
# Attention Score Functions

- Dot-product:  $\text{score}(h_t, s_k) = h_t^T s_k$

$$\begin{matrix} h_t^T \\ \text{---} \end{matrix} \times \begin{matrix} s_k \\ | \\ \text{---} \end{matrix}$$

- Bilinear:  $\text{score}(h_t, s_k) = h_t^T W s_k$

$$\begin{matrix} h_t^T \\ \text{---} \end{matrix} \times \begin{matrix} W \\ | \\ \text{---} \end{matrix} \times \begin{matrix} s_k \\ | \\ \text{---} \end{matrix}$$



# Attention Score Functions

- Dot-product:  $\text{score}(h_t, s_k) = h_t^T s_k$

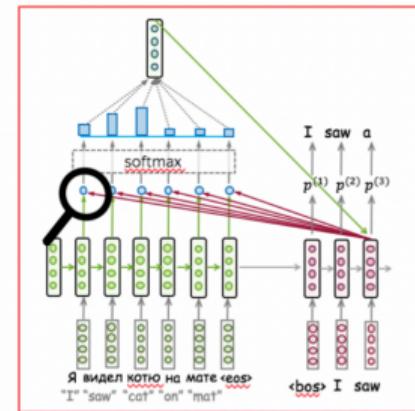
$$h_t^T \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} s_k$$

- Bilinear:  $\text{score}(h_t, s_k) = h_t^T W s_k$

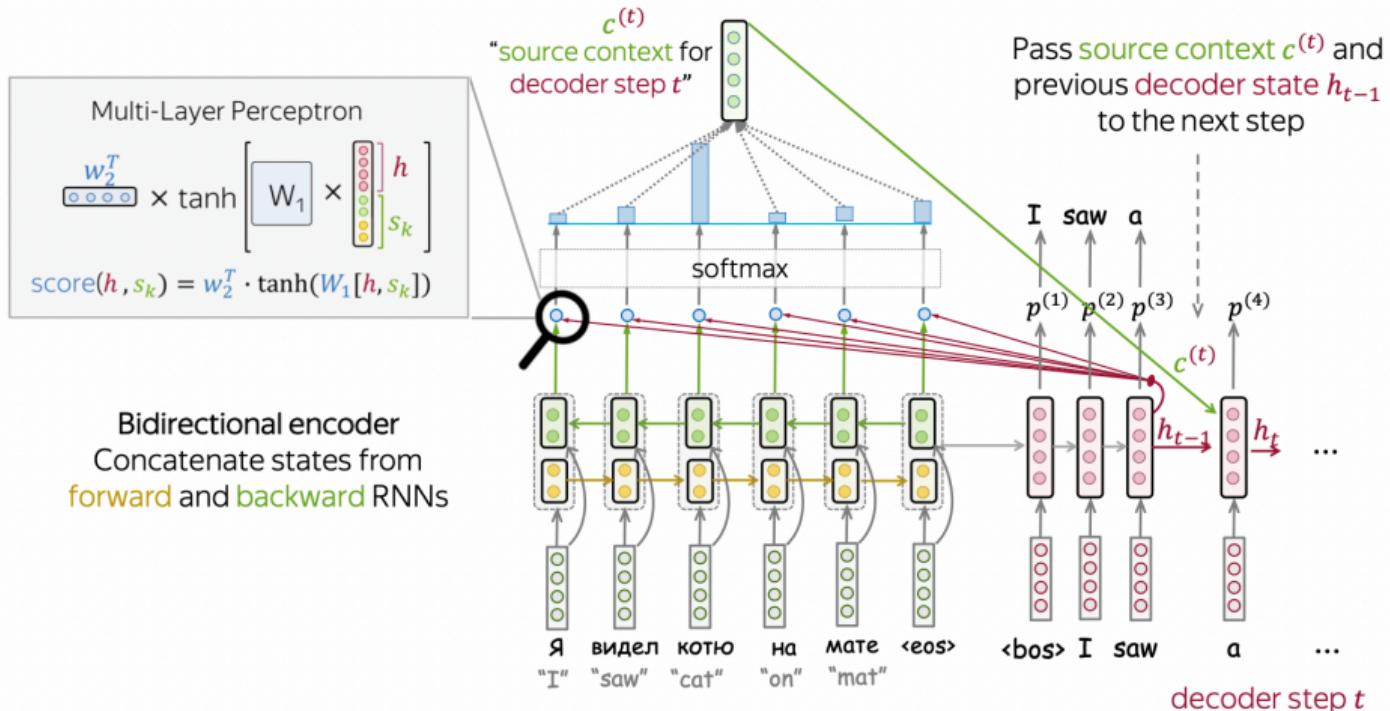
$$h_t^T \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} W \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} s_k$$

- Multi-Layer Perceptron:  $\text{score}(h_t, s_k) = w_2^T \cdot \tanh(W_1[h_t, s_k])$

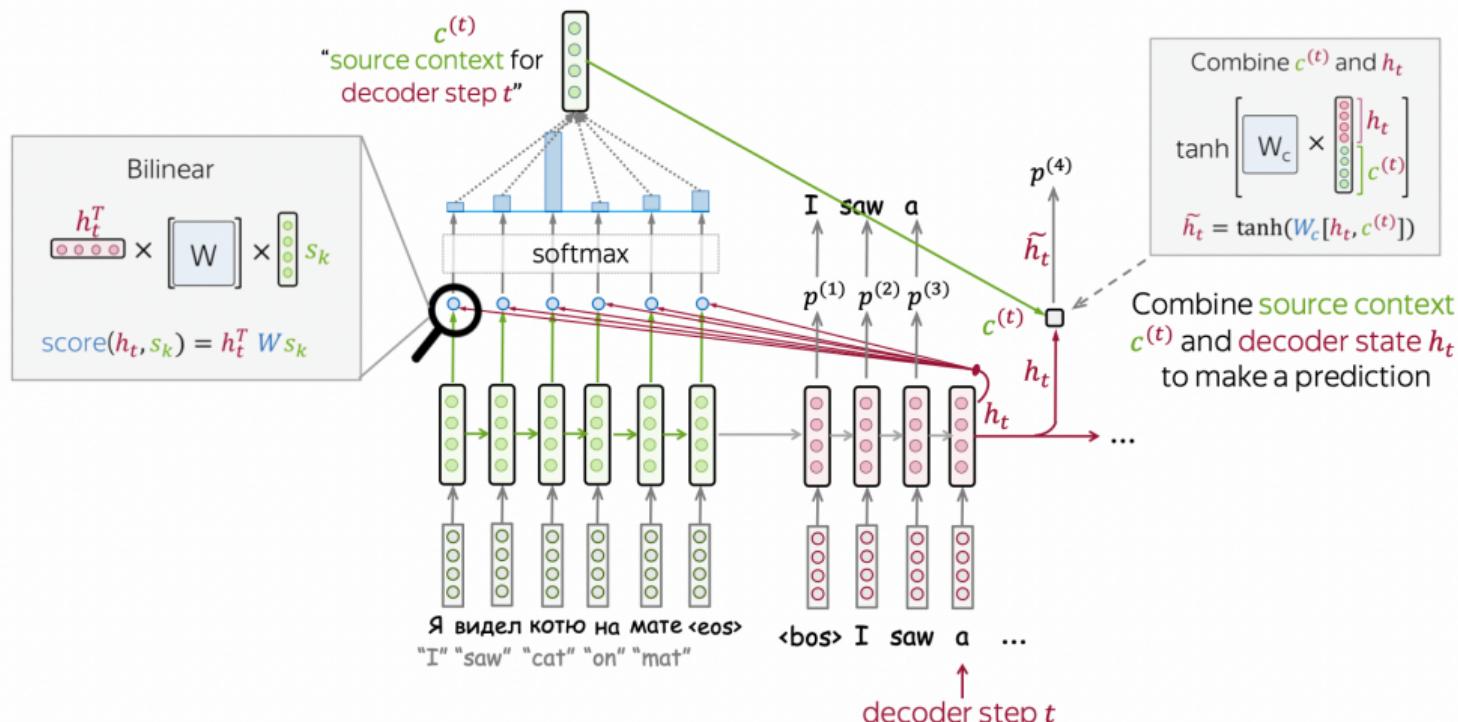
$$w_2^T \times \tanh \left[ \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} W_1 \times \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} h_t \right] \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} s_k$$



# Bahdanau Model (the original attention model)



# Luong Model



# Attention

The  
agreement  
on  
the  
European  
Economic  
Area  
was  
signed  
in  
August:  
1992  
. <end>

This heatmap visualizes attention weights for the English sentence. The matrix is 10x10, corresponding to the tokens: The, agreement, on, the, European, Economic, Area, was, signed, in, August:, 1992, ., <end>. The highest attention values are concentrated along the diagonal, indicating strong self-attention for each token. There are also secondary clusters of higher attention values, particularly for 'European' and 'Economic'.

L'  
accord  
sur  
la  
zone  
économique  
européenne  
a  
été  
signé  
en  
août  
1992  
. It  
should  
be  
noted  
that  
the  
marine  
environment  
is  
the  
least  
known  
of  
environments  
. <end>

This heatmap visualizes attention weights for the French sentence. The matrix is 10x10, corresponding to the tokens: L', accord, sur, la, zone, économique, européenne, a, été, signé, en, août, 1992, ., It, should, be, noted, that, the, marine, environment, is, the, least, known, of, environments, ., <end>. Similar to the English case, attention is primarily along the diagonal, with notable off-diagonal patterns for words like 'zone', 'économique', 'européenne', 'marine', and 'environments'.