

Natural Language Processing

Лекция 1. Recap.

Эль-Айясс Дани Валид

МФТИ

16 сентября 2023

О себе

Дани Эль-Айясс:

- Магистр по направлению «Прикладная математика и информатика», ВМК МГУ (кафедра ММП)
- Исполнительный директор в SberDevices, разрабатываю GigaChat

Контакты:

- Почта: dayyass@yandex.ru
- Телеграм: @dayyass
- LinkedIn: <https://www.linkedin.com/in/dayyass/>
- GitHub: <https://github.com/dayyass>

План курса

1. Recap
2. Большие языковые модели
3. Alignment: инструктивное дообучение и RLHF
4. Суммаризация текстов и вопросно-ответные системы
5. Информационный поиск
6. Мультимодальная обработка текстов

План

- Что такое NLP?
 - Задачи NLP
- Векторные представления
 - One-Hot Encoding
 - Word2Vec
 - FastText
- RNN
 - Classification
 - Token Classification
 - Language Modeling
- Seq2Seq
 - Machine Translation
 - Attention
- Transformer
 - Subword Tokenization
 - GPT
 - BERT

Что такое NLP?

Текстовые данные

- Большая часть данных в мире представлена в текстовом виде
- Текстовые данные могут быть:
 - структурированными (графы знаний, базы данных)
 - неструктурированными (сырые тексты)
 - частично структурированными (JSON, XML)

Естественный язык

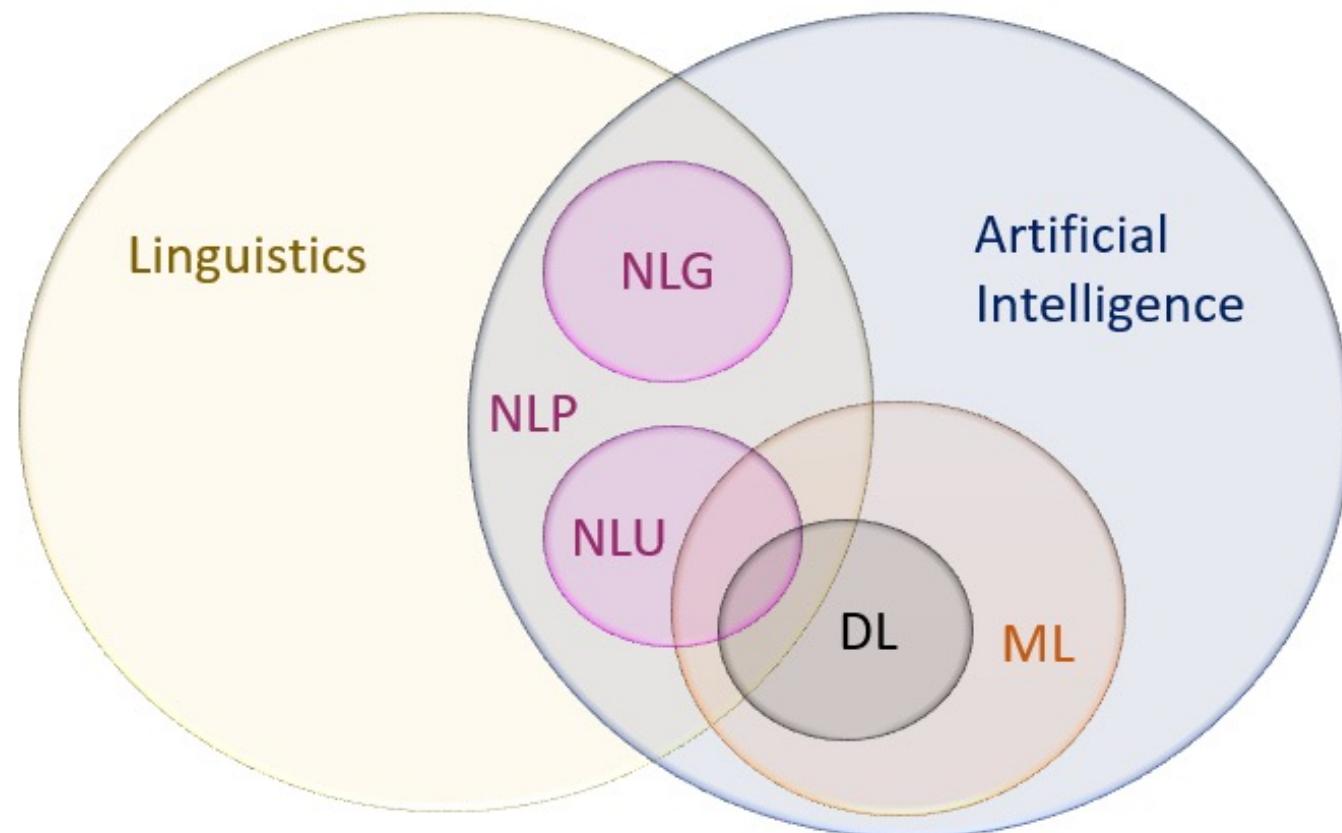
- Естественный язык – способ общения между людьми
- Можем противопоставить его *формальным и искусственным языкам*:
 - Языки программирования – Programming Language Processing (PLP)
- «Глокая куздра штеко будланула бокра и кудрячит бокренка» (Л.В. Щерба, 1930-е)

Правила языка

- Выстраивается некоторая иерархия:
 - Графематические – как разделять слова и предложения между собой
 - Морфологические – как строить и изменять слова
 - Синтаксические – как согласовывать словоформы друг с другом
 - Семантические – как применять все предыдущие правила, чтобы сообщить необходимую информацию
 - Прагматические / стилистические – «уместность» словоупотребления в конкретной ситуации

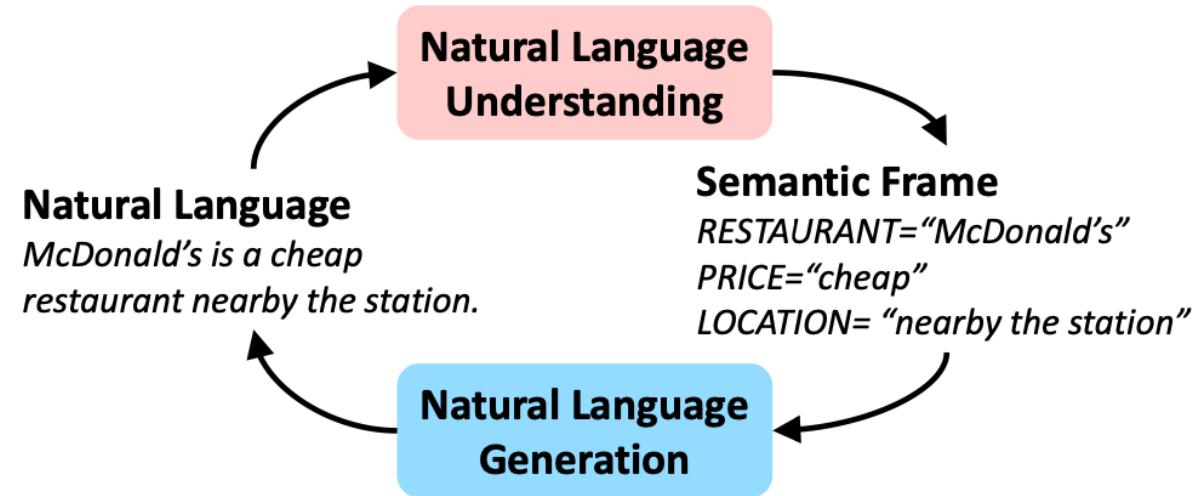
Обработка естественного языка (NLP)

- Положение NLP среди наук по анализу и обработке данных:



Структура NLP

- Внутри NLP условно выделяются два направления:
 - понимание языка (NLU)
 - генерация языка (NLG)
- Текст → NLU → смысл → NLG → текст



Пирамида NLP



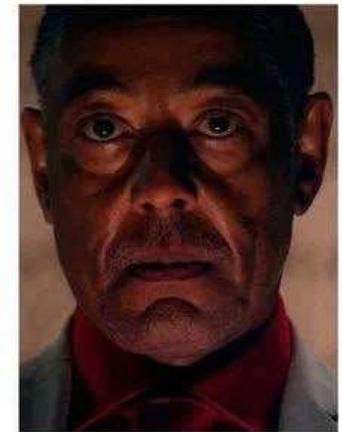
P.S. В самом низу графематический уровень

Особенности NLP

- Базовая структурная единица языка — слово
 - Даже вне контекста оно несет полезную информацию
 - У слов есть различные словоформы в зависимости от контекста
 - Многозначность слова (полисемия, омонимия)
- Текст без дополнительной разметки имеет внутреннюю структуру

РУССКИЕ ТАКИЕ:

**НЕ НАДО МЕНЯ
УГОВАРИВАТЬ**



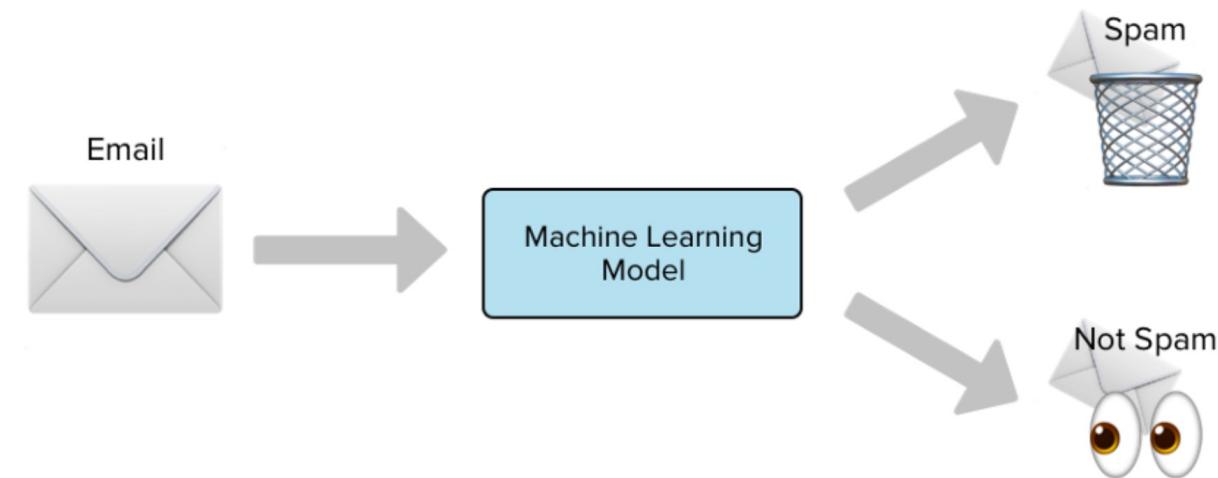
**МЕНЯ НЕ НАДО
УГОВАРИВАТЬ**



Задачи NLP

Задача классификации текстов

- Задачи NLP можно формулировать с технической и продуктовой точек зрения.
- Классификация - одна из основных задач в NLP, лежит в основе многих продуктовых задач:
 - Анализ тональности
 - **Фильтрация спама**
 - Определение намерений
 - Категоризация новостей и статей



Задача разметки последовательностей

- Извлечение информации
 - Распознавание именованных сущностей
- Частеречная разметка
- Разрешение кореференции

contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported ORG by F.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top F.B.I. GPE counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump PERSON were uncovered, was fired. CreditT.J. Kirkpatrick PERSON for The New York TimesBy Adam Goldman ORG and Michael S. SchmidtAug PERSON . 13 CARDINAL , 2018WASHINGTON CARDINAL — Peter Strzok PERSON , the F.B.I. GPE senior counterintelligence agent who disparaged President Trump PERSON in inflammatory text messages and helped oversee the Hillary Clinton PERSON email and Russia GPE investigations, has been fired for violating bureau policies, Mr. Strzok PERSON 's lawyer said Monday DATE . Mr. Trump and his allies seized on the texts — exchanged during the 2016 DATE campaign with a former F.B.I. GPE lawyer, Lisa Page — in PERSON assailing the Russia GPE investigation as an illegitimate "witch hunt." Mr. Strzok PERSON , who rose over 20 years DATE at the F.B.I. GPE to become one of its most experienced counterintelligence agents, was a key figure in the early months DATE of the inquiry. Along with writing the texts, Mr. Strzok PERSON was accused of sending a highly sensitive search warrant to his personal email account. The F.B.I. GPE had been under immense political pressure by Mr. Trump PERSON to dismiss Mr. Strzok PERSON , who was removed last summer DATE from the staff of the special counsel, Robert S. Mueller III PERSON . The president has repeatedly denounced Mr. Strzok PERSON in posts on

Задача машинного перевода

- Одна из фундаментальных задач NLP, двигатель многих исследований и открытий:
 - Attention
 - Transformers
- Машинный перевод:
 - Статистический
 - Нейронный

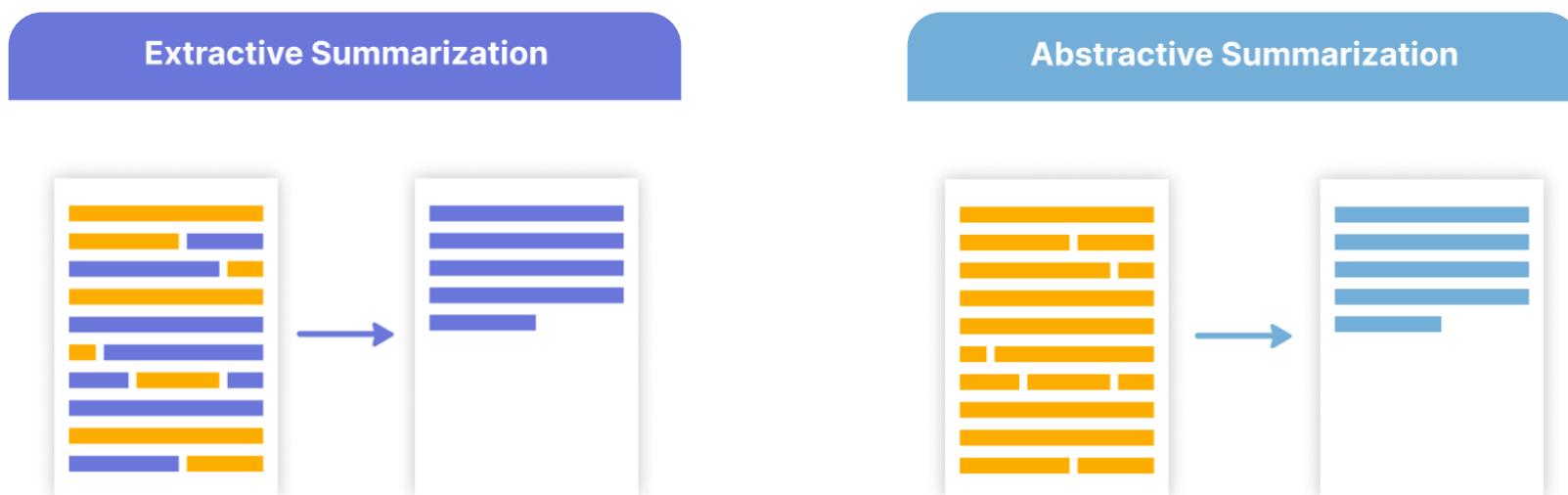
it is raining cats and dogs

✗ идет дождь из кошек и собак

✓ льет как из ведра

Задача суммаризации текстов

- Для текстового документа нужно сгенерировать краткое изложение
- Важна не только передача смысла, но и сохранение важных фактов



Задача поиска ответов на вопросы

- Вопросно-ответные системы (QA-система) используются для поиска ответов на вопросы, заданные на естественном языке
- QA-системы часто используются в качестве элементов поисковых систем
- Пример: «что такое луна?»



Луна

Естественный спутник



Луна — единственный естественный спутник Земли. Самый близкий к Солнцу спутник планеты, так как у ближайших к Солнцу планет их нет. Второй по яркости объект на земном небосводе после Солнца и пятый по величине естественный спутник планеты Солнечной системы. Среднее расстояние между центрами Земли и Луны — 384 467 км. [Википедия](#)

Расстояние до Земли: 384 400 км

Ускорение свободного падения: 1,62 м/с²

Радиус: 1 737,1 км

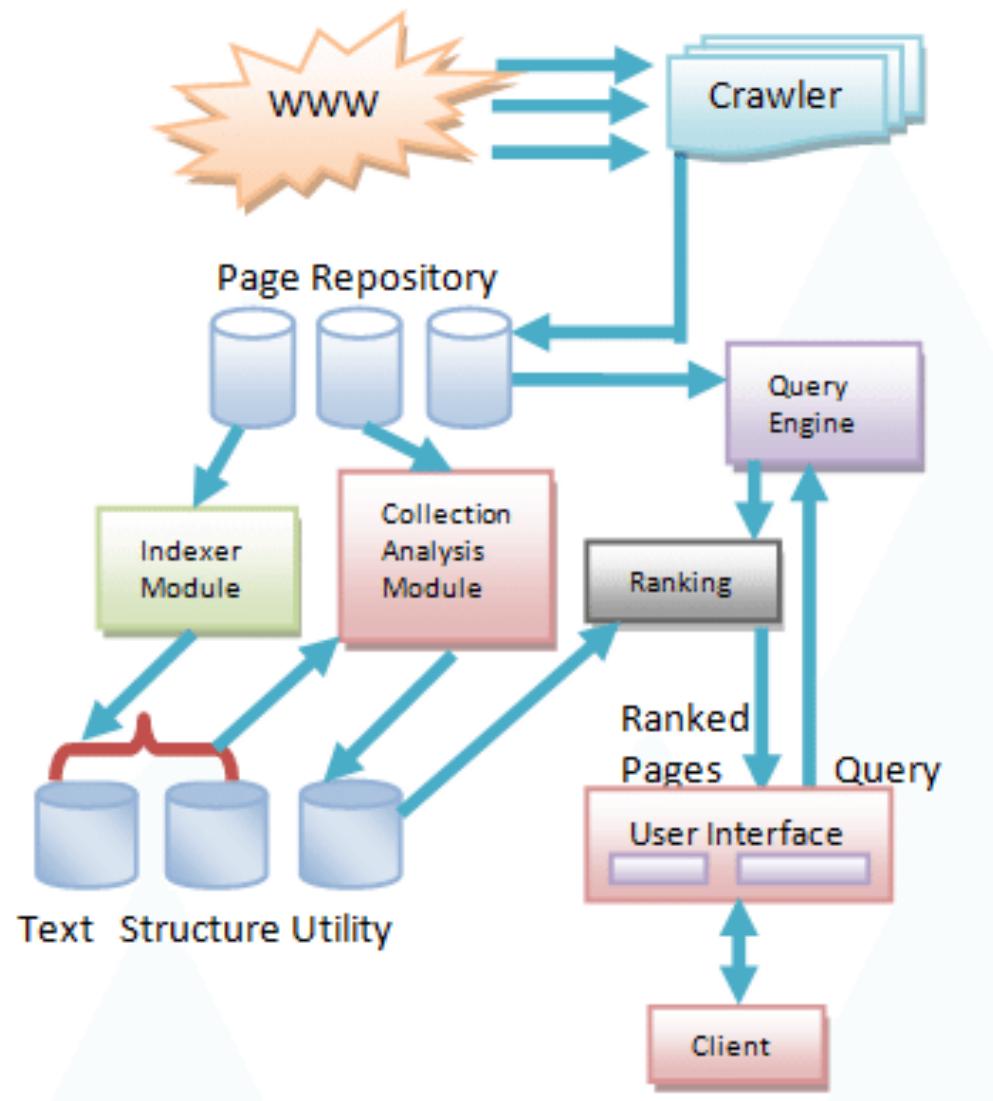
Возраст: 4,53E9 лет

Период обращения: 27 дней

Обращается вокруг: Земля

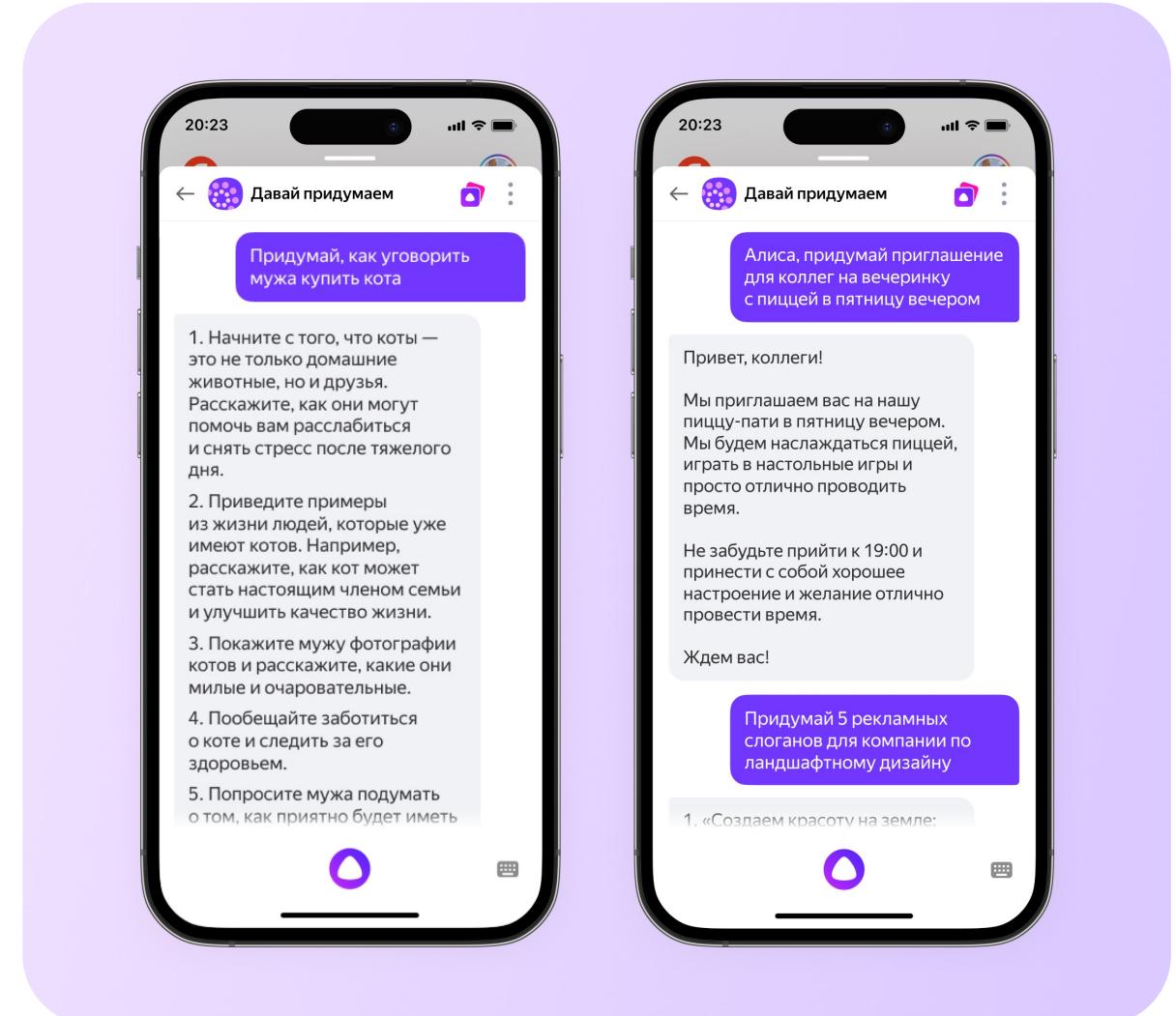
Задача ранжирования

- Ранжирование решает задачу сортировки объектов по заданному критерию полезности:
 - *информационный поиск* — релевантность страницы сайта пользовательскому запросу
 - *рекомендации* — близость текстовой статьи к текущим интересам пользователя



Задача ведения диалога

- Диалоговые системы (чат-боты) общаются с человеком на естественном языке
- Хороший пример NLU -> NLG



Этапы решения NLP задач

Этапы решения NLP-задачи

Всё так же, как и при обработке других типов данных:

- Выбор верной метрики качества
- Сбор обучающих и тестовых данных
- *Предобработка данных*
- *Формирование признакового описания текста*
- Выбор подхода и класса моделей
- Обучение моделей и настройка решения

Предобработка текстов

Базовые шаги предобработки:

1. токенизация
2. приведение к нижнему регистру
3. удаление пунктуации
4. удаление стоп-слов
5. фильтрация слов по частоте/длине/регулярному выражению
6. нормализация слов - лемматизация или стемминг

Признаковые описания документов

- Рассмотрим задачу классификации
- Обычно в ML данные представляют собой матрицу «объекты-признаки»:

Номер автомобиль	Тип топлива	Мощность двигателя	...	Масса
1	Бензин	120	...	1700
...
N	Дизель	160	...	2100

- Для текстов тоже нужно как-то получить такую матрицу

Модель мешка слов

- Можно проверять наличие всех возможных слов из некоторых словаря:

Номер текста	Содержит «абрикос»	...	Содержит «яблоко»
1	0	...	1
...
N	1	...	0

- Пусть значением признака будет не наличие слова, а число его вхождений в документ («мешок слов»):

Номер текста	Вхождений «абрикос»	...	Вхождений «яблоко»
1	0	...	23
...
N	2	...	0

TF-IDF

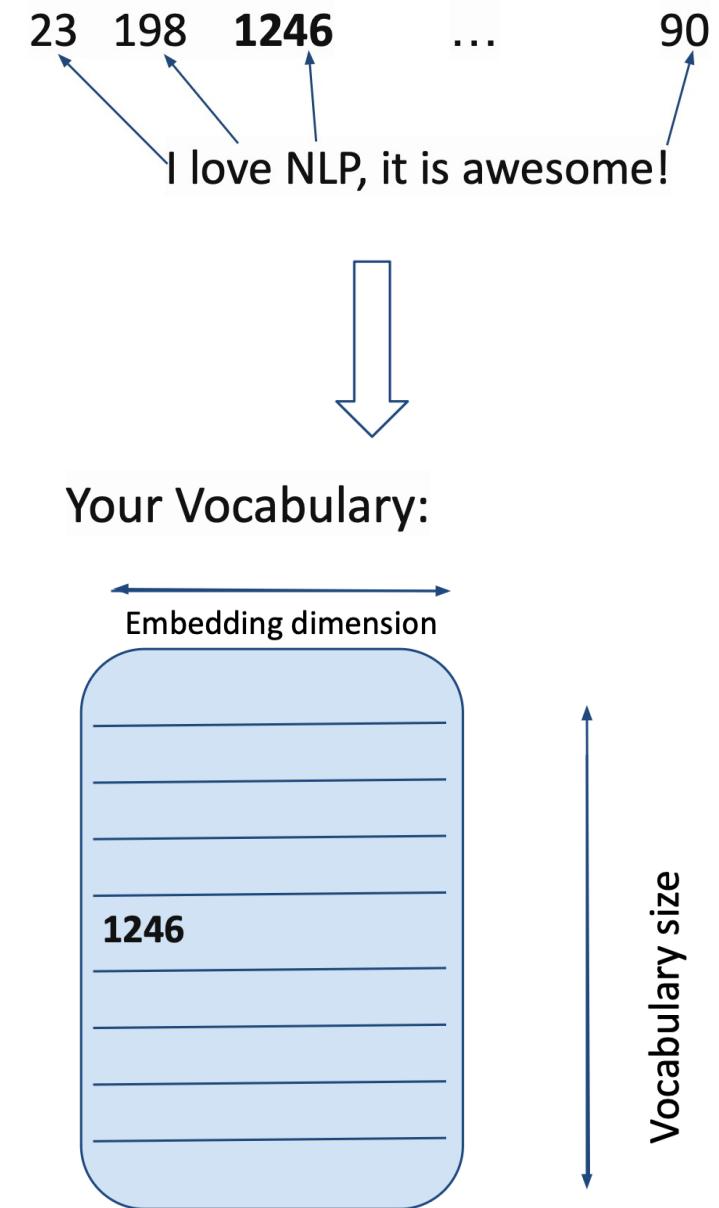
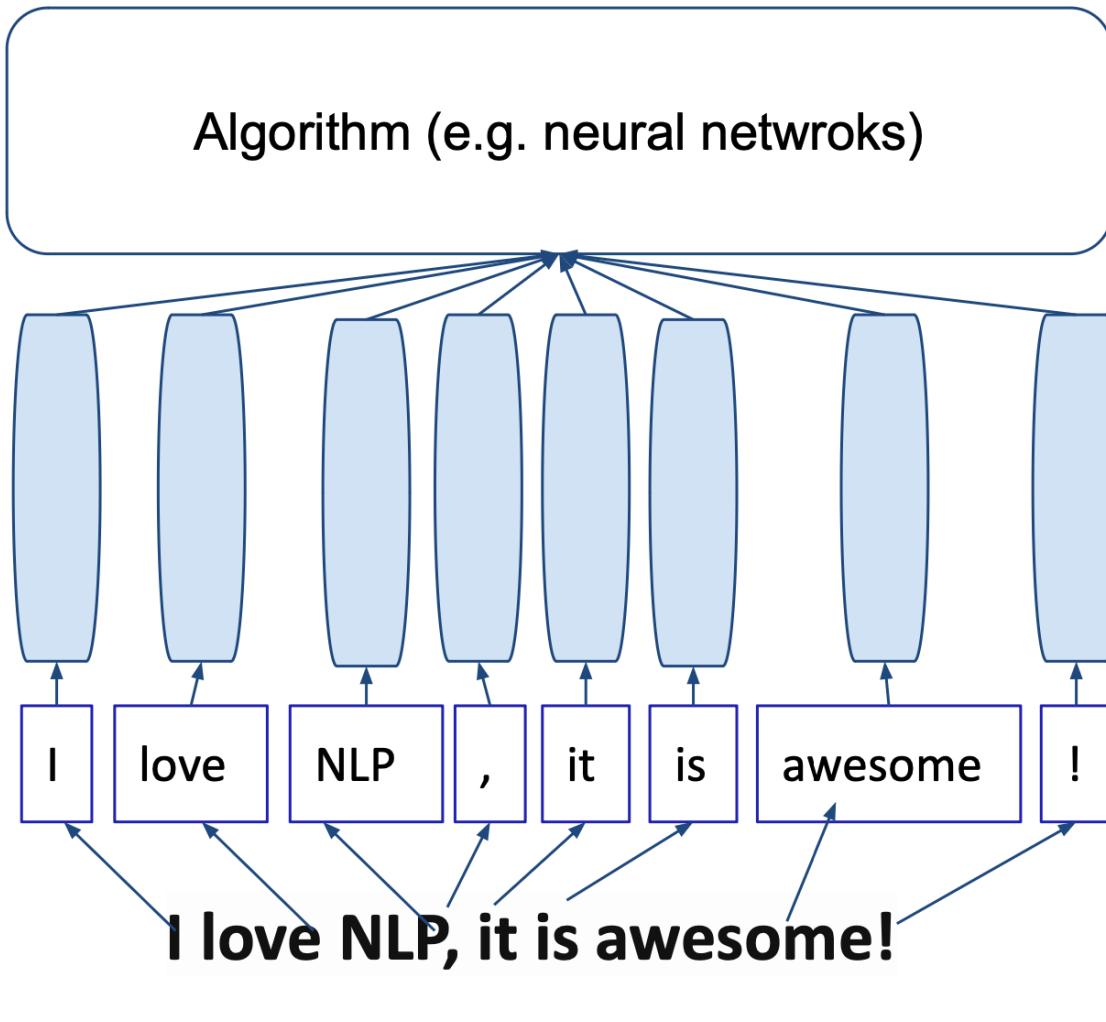
- Представление «мешка слов» часто используется при обработке текстов, но частота встречаемости слов не самый информативный признак
- Идея: хотим выделить слова, которые часто встречаются в данном тексте, и редко — в других текстах - используем значения TF-IDF:

$$v_{wd} = tf_{wd} \times \log \frac{N}{df_w}$$

- ▶ tf_{wd} — доля слова w в словах документа d
- ▶ df_w — число документов, содержащих w
- ▶ N — общее число документов

Векторные представления

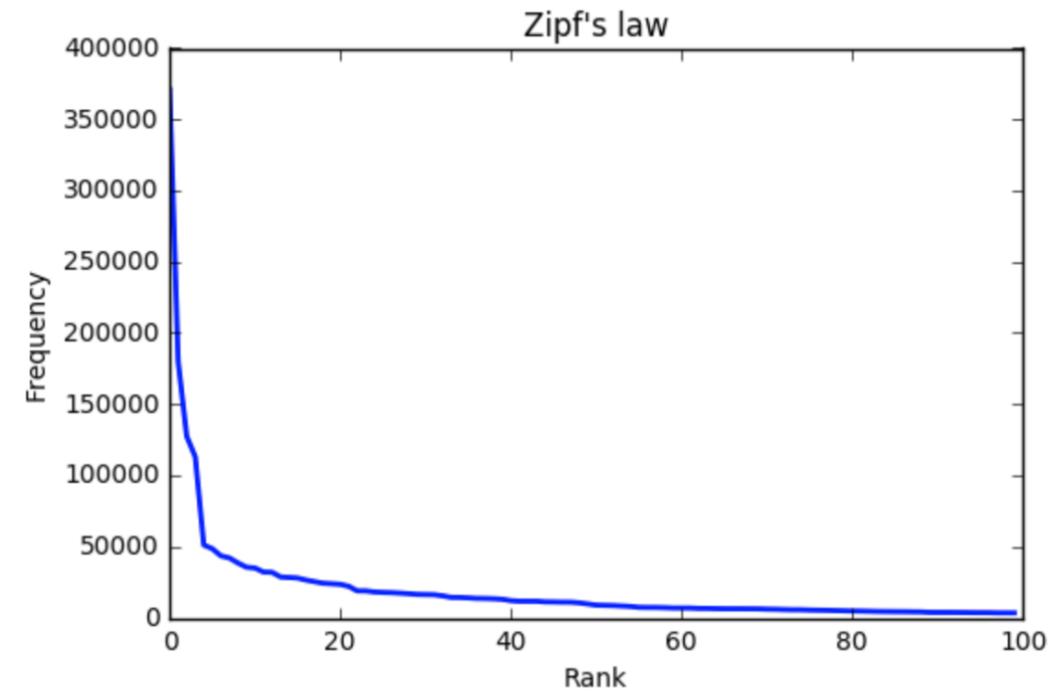
Look-up Table



One-Hot Encoding

Самый простой способ кодирования категориальных признаков:

"a"	"abbreviations"	"zoology"	"zoom"
1	0	0	0
0	1	0	1
0	0	0	0
.	.	.	.
.	.	.	.
.	.	0	0
0	0	0	0
0	0	1	0
0	0	0	1



А что с TF-IDF?

<https://medium.com/m/global-identity-2?redirectUrl=https%3A%2F%2Fyearofai.com%2Flenny-2-autoencoders-and-word-embeddings-oh-my-576403b0113a>

One-Hot Encoding

Плюсы :

- легко и быстро построить
- неплохое качество решения задач на длинных текстах

Минусы  :

- большая размерность
- разреженность
- ортогональность
- отсутствие информации о словах:
 - нет близости среди векторов
 - нет «смысла» (meaning)
- out-of-vocabulary

Построение векторных представлений

Дано:

- $D = \{w_1, w_2, \dots, w_N\}$ — текстовая коллекция
- D — конкатенация всех документов
- $w_i \in W$ — слово, W — словарь коллекции

Найти: векторное представление $v_w \in R^m$ для каждого слова w

Какие представления считать хорошими?

- Близким по смыслу словам соответствуют близкие по расстоянию вектора
- Небольшая размерность — $m \ll |W|$
- Интерпретируемые арифметические операции в пространстве R^m
- Качество решения конечной задачи

Дистрибутивная семантика

Что такое троллингер (trollinger)?

Дистрибутивная семантика

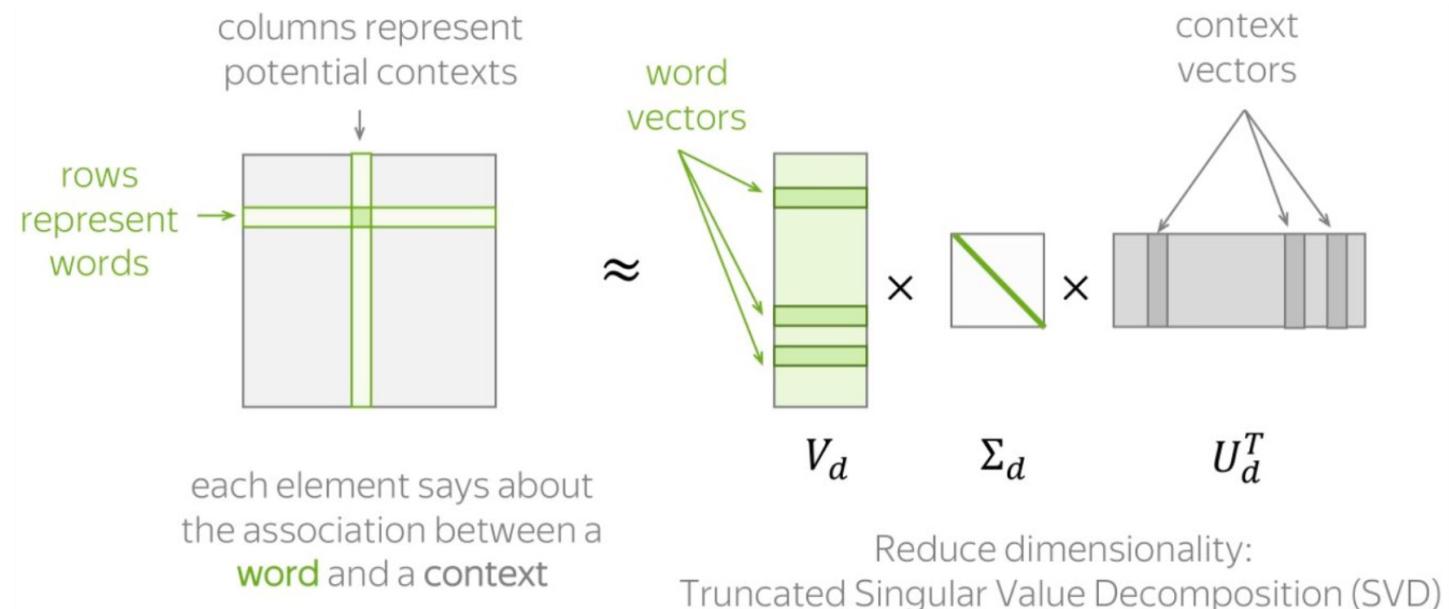
А если добавить контекст?

1. Не садись за руль после _____
2. В Италии _____ известен под именами вернач и скьява
3. Бутылка _____ стояла на столе
4. Сорт винограда _____ а также одноименная марка вина, выращивается в немецком регионе Вюртемберг

	(1)	(2)	(3)	(4)	...
троллингер	1	1	1	1	
масло	0	0	1	0	
вино	1	0	1	0	
пятница	0	0	0	0	

Count-based методы

- Построить *word-context matrix* по нашему словарю
 - Нужно определить, что такое контекст
 - Придумать, как считать элемент матрицы (близость слова и контекста)
- Понизить размерность



Матрица совстречаемостей (Co-occurrence matrix)

$X \in R^{|W| \times |W|}$ — матрица совстречаемостей, $X_{wc} = f(w, c, D)$

1. $X_{wc} = n_{wc}$ — количество совстречаний слов w и c
2. $X_{wc} = \text{PMI}(w, c)$ — pointwise mutual information

$$\text{PMI}(w, c) = \log \frac{p(w, c)}{p(w)p(c)} = \log \frac{n_{wc}}{n_c n_w} + \text{Const}$$

n_w — число появлений слова w в коллекции

3. $X_{wc} = \text{PPMI}(w, c) = \max(0, \text{PMI}(w, c))$
4. TF-IDF - ???

Матрица совстречаемостей (Co-occurrence matrix)

Возможны различные варианты учета со-встречаемости слов:

1. сумма по всей коллекции числа попаданий пары слов в окно фиксированного размера
2. количество документов, хоть раз содержащих пару слов
3. количество документов, хоть раз содержащих пару слов в окне

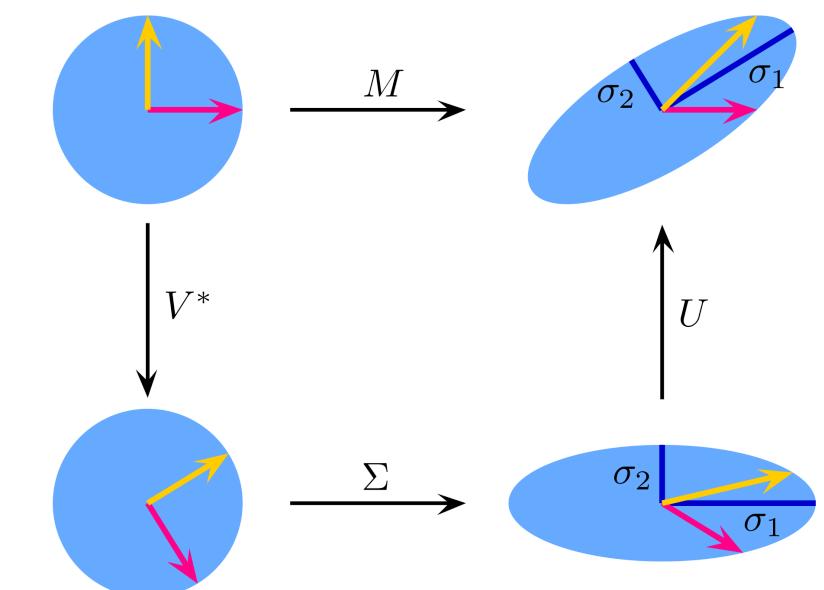
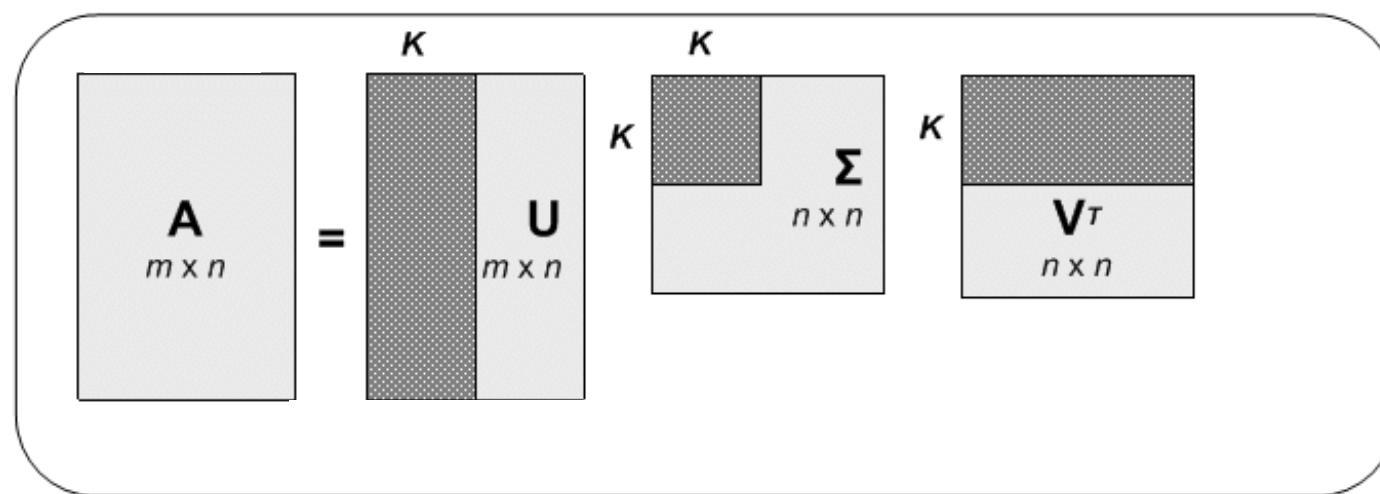
X_w — эмбеддинг $\in R^{|W|}$, решающий проблему ортогональности.

Как получить эмбеддинг $\in R_m$, $m \ll |W|$?

SVD для построения представлений

SVD-разложение: $X = USV^T$

Из столбцов матрицы U выбираются первые K компонент



Есть ли в SVD какая либо оптимизация?

$$M = U \cdot \Sigma \cdot V^*$$

Недостатки SVD

1. Относительно низкое качество получаемых представлений¹
2. Сложность работы с очень большой и разреженной матрицей
3. Сложность добавления новых слов/документов (решается инкрементальными методами построения)

1. При определенных условиях показывает хорошее качество на стандартных бенчмарках:
Levy et al (ACL 2015), Improving Distributional Similarity with Lessons Learned from Word Embeddings.

GloVe

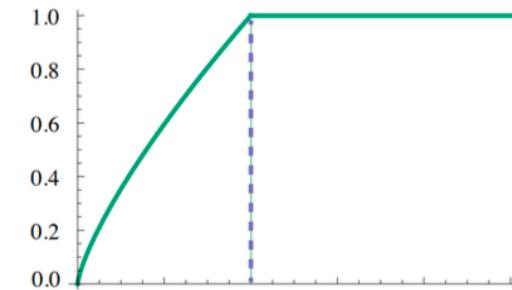
- GloVe — ещё одно матричное разложение.

- Методом Adagrad обучается функционал:

$$\mathcal{L} = \sum_{w \in W} \sum_{c \in W} F(n_{wc}) (\langle u_w, v_c \rangle + b_w + \hat{b}_c - \log n_{wc})^2 \longrightarrow \min_{U, V, b, \hat{b}}$$

- Боремся с шумовыми редкими словами с помощью F :

$$F(n_{wc}) = \begin{cases} \left(\frac{n_{wc}}{n_{max}}\right)^{3/4}, & n_{wc} < n_{max} \\ 1, & \text{иначе} \end{cases}$$



Тематическое моделирование

- Пример разложения с не квадратной матрицей

$$F = p(w|d) \quad \Phi = p(w|t)$$
$$\Theta = p(t|d)$$

The diagram illustrates the decomposition of a word-document matrix F into two matrices Φ and Θ . The matrix F has 8 words (rows) and 5 documents (columns), represented by a green grid. It is decomposed into $\Phi \times \Theta$, where Φ has 8 words (rows) and 3 topics (columns), and Θ has 3 topics (rows) and 5 documents (columns). The matrix Φ is shown with colored blocks along its diagonal, indicating non-zero entries for each word-topic pair. The matrix Θ is shown with colored blocks in its last column, indicating non-zero entries for each topic-document pair.

Count-based методы

Плюсы :

- неплохое качество в некоторых задачах
- маленькая размерность
- близким словам соответствуют близкие вектора

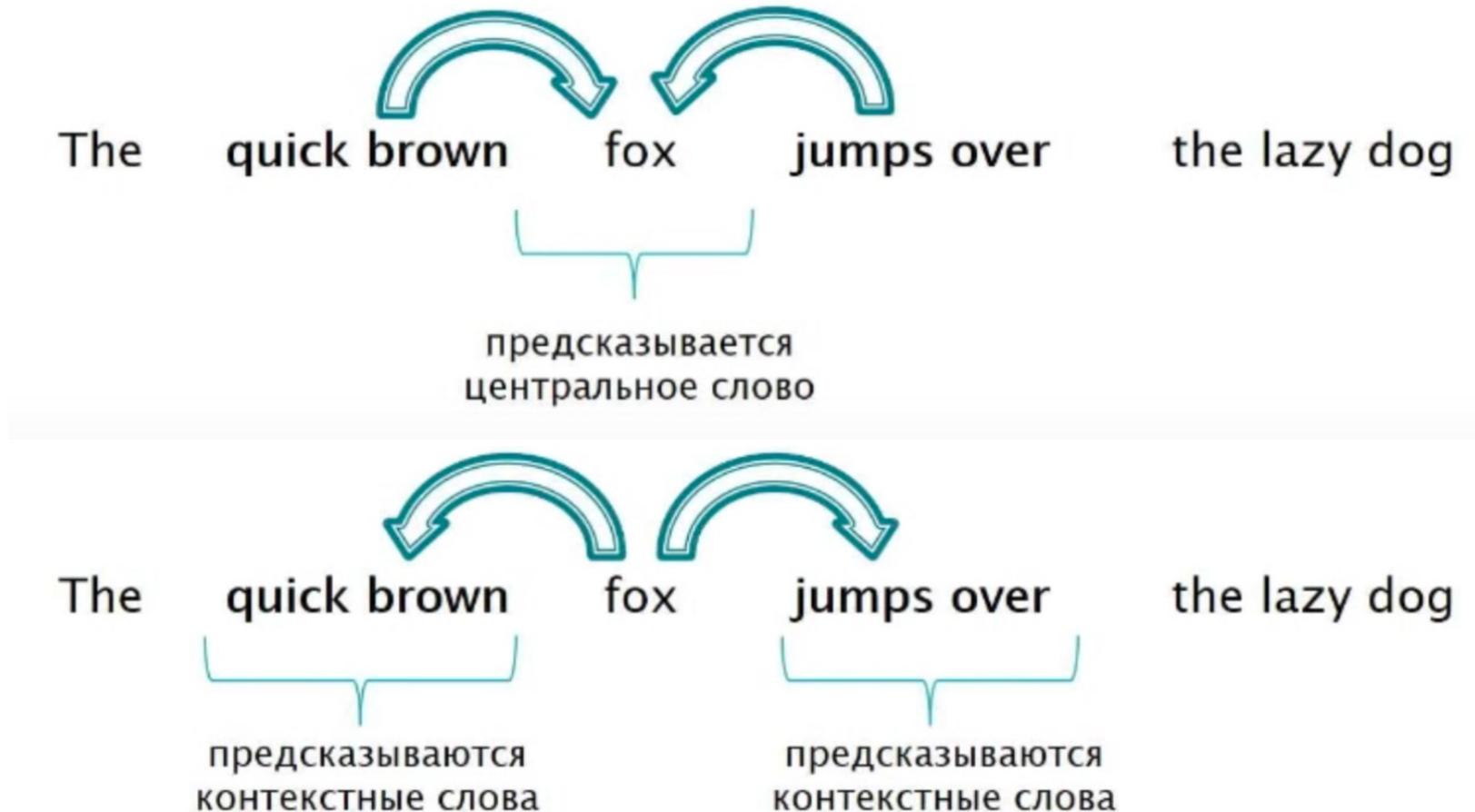
Минусы  :

- нет хорошего механизма обработки новых слов на teste (out-of-vocabulary)
- необходимо собирать огромную (но разреженную!) матрицу совстречаемостей для обучения

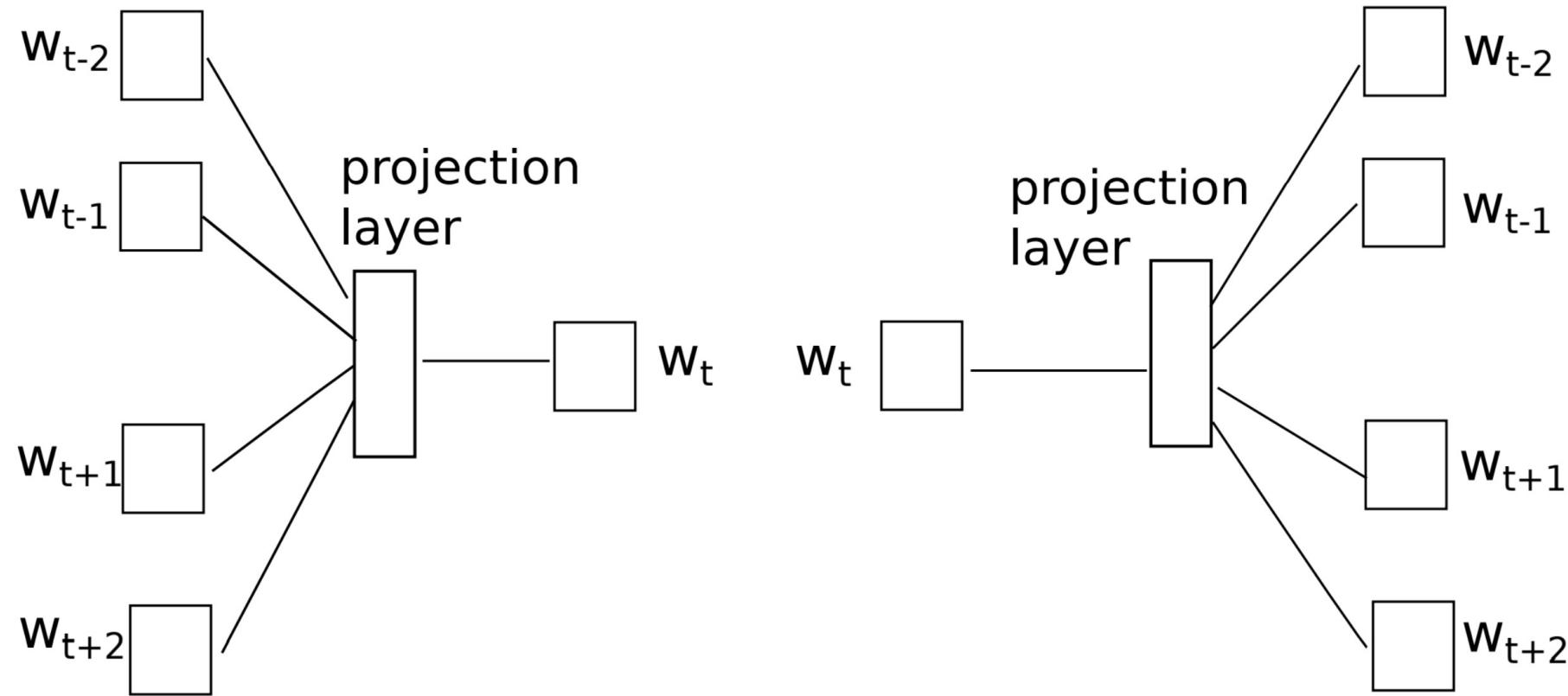
Prediction-based методы

- Хотим обновлять параметры модели «на ходу», не составляя матрицу совстречаемостей.
- **Идея.** Обучаем модель «воспроизводить» контекст:
 - Модель Continuous BOW (CBOW) — по словам контекста необходимо предсказать центральное слово
 - Модель Skip-gram — по центральному слову, необходимо предсказать каждое из слов контекста
- **Обратите внимание!** Идея очень схожа с языковой моделью, но контекст — не только слова перед словом.

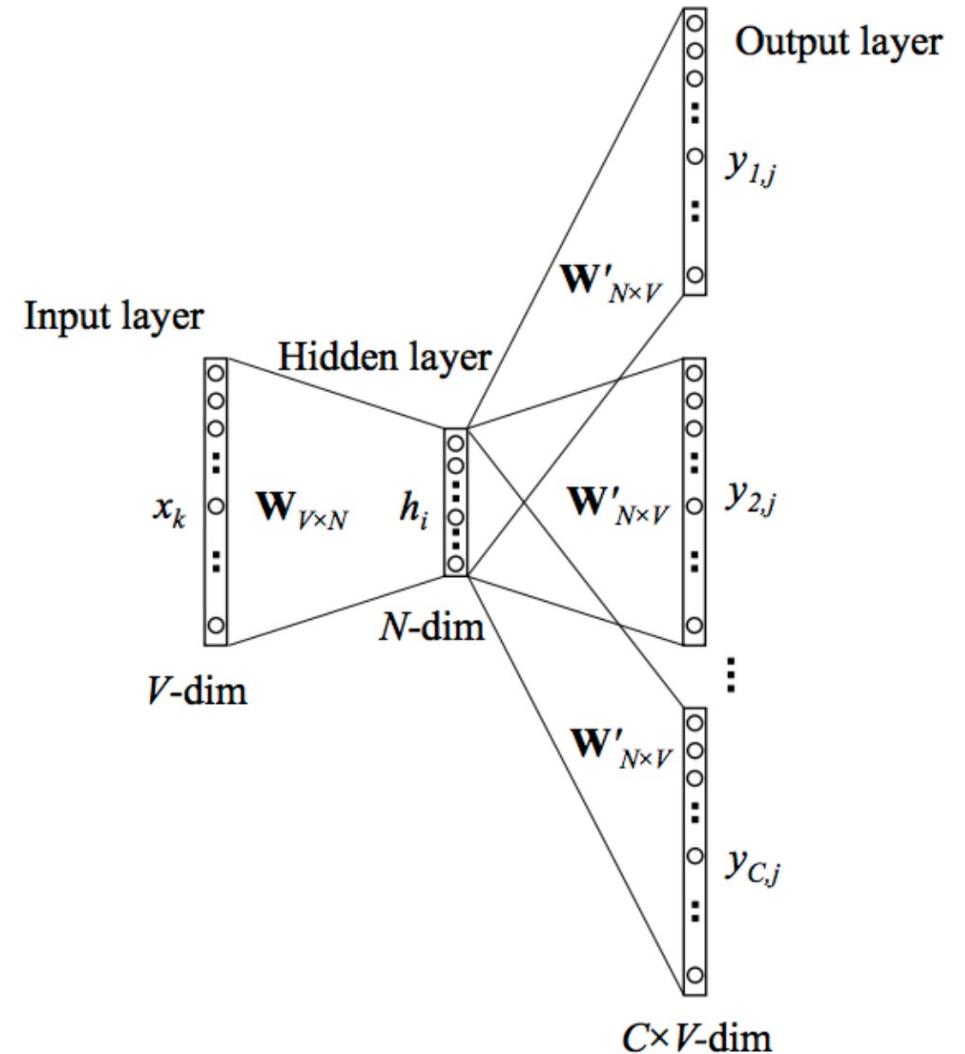
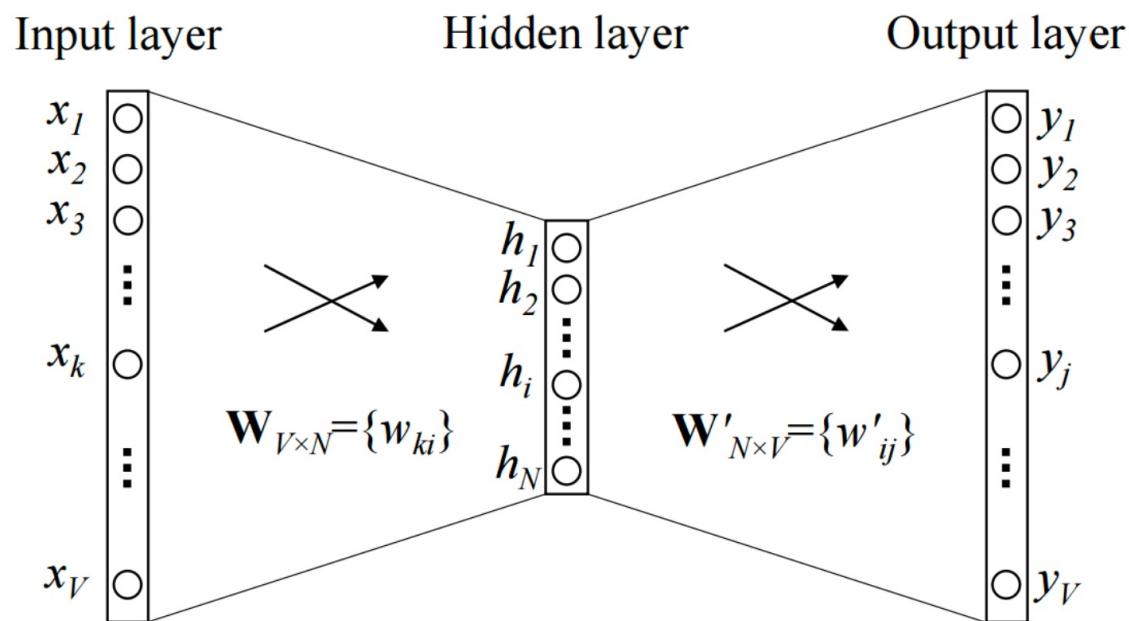
Модель CBOW и Skip-gram



Модель CBOW и Skip-gram



Модель CBOW и Skip-gram



Модель CBOW

- Функционал обучения:

$$\sum_{i=1}^N \log p(w_i | C(i)) \rightarrow \max_{V, U}$$

$C(i) = \{w_{i-k}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+k}\}$ — локальный контекст w_i

- 1 этап — вычисление среднего входных векторов
- 2 этап — применение линейного слоя с softmax активацией:

Модель Skip-gram

- Функционал обучения:

$$\sum_{i=1}^N \sum_{w \in C(i)} \log p(w|w_i) = \sum_{i=1}^N \sum_{\substack{j=-k \\ j \neq 0}}^k \log p(w_{i+j}|w_i) \rightarrow \max_{V,U}$$

$$p(w|w_i) = \underset{w \in W}{\text{softmax}} U v_{w_i} = \underset{w \in W}{\text{softmax}} \langle u_w, v_{w_i} \rangle$$

- CBOW и Skip-gram обучаются с помощью SGD
- Skip-gram лучше моделирует редкие слова коллекции
- Какая сложность итерации обучения CBOW и Skip-gram?

Способы ускорения модели

1. Замена softmax на другую функцию, задающую распределение:
 - **Hierarchical softmax¹**
 - Differentiated softmax
 - ...
2. Замена функционала модели на более простой:
 - Noise contrastive estimation
 - **Negative sampling²**
 - Importance sampling
 - Self-normalization
 - Infrequent Normalization
 - ...

1. Mikolov (NIPS 2013), Distributed representations of words and phrases and their compositionality

2. Ruder; On word embeddings - Part 2: Approximating the Softmax; <http://ruder.io/word-embeddings-softmax/>

Negative sampling (сэмплирование негативных примеров)

- Исходный метод: вероятность встретить w в контексте с c в коллекции $|W|$ вероятностных распределений, каждое с $|W|$ исходами
- **Negative sampling:** вероятность встретить пару (w, c) в коллекции $|W| \times |W|$ вероятностных распределений, каждое с 2 исходами

$$p(1|c, w) = \sigma(\langle v_c, u_w \rangle) = 1 - p(0|c, w)$$

- В чем проблема этой модели?

$$\sum_{i=1}^N \sum_{w \in C(i)} \log p(1|w, w_i) \rightarrow \max_{V, U}$$

Negative sampling (сэмплирование негативных примеров)

- Чтобы не переобучаться, будем на каждой итерации сэмплировать n случайных негативных примеров:

$$\sum_{i=1}^N \left(\sum_{w \in C(i)} \log p(1|w_{i+j}, w_i) + \sum_{w'_k \sim p(w)^{3/4}} \log p(0|w_i, w'_k) \right) \rightarrow \max_{V, U}$$

- Важно.** Прием популярен не только при обучении skip-gram, но и в любой ситуации, когда у вас в выборке только позитивные пары.

Итог по word2vec

Плюсы :

- Хорошее качество в самых разных прикладных задачах
- Маленькая размерность
- Близким словам соответствуют близкие вектора

Минусы :

- Плохой механизм обработки новых слов на тесте
- Требуют большего корпуса чем count-based модели

FastText

- FastText¹ — построение представлений слов как суммы представлений для буквенных n-грамм слова.

В Skip-gram меняется только подсчёт вектора u_w :

$$u_w = \sum_{g \in G(w)} u_g, \quad G(w) — n\text{-граммы слова } w$$

Пример: $G(\text{where}) = \text{_wh} + \text{whe} + \text{her} + \text{ere} + \text{re_}$

Итог по fasttext

Плюсы :

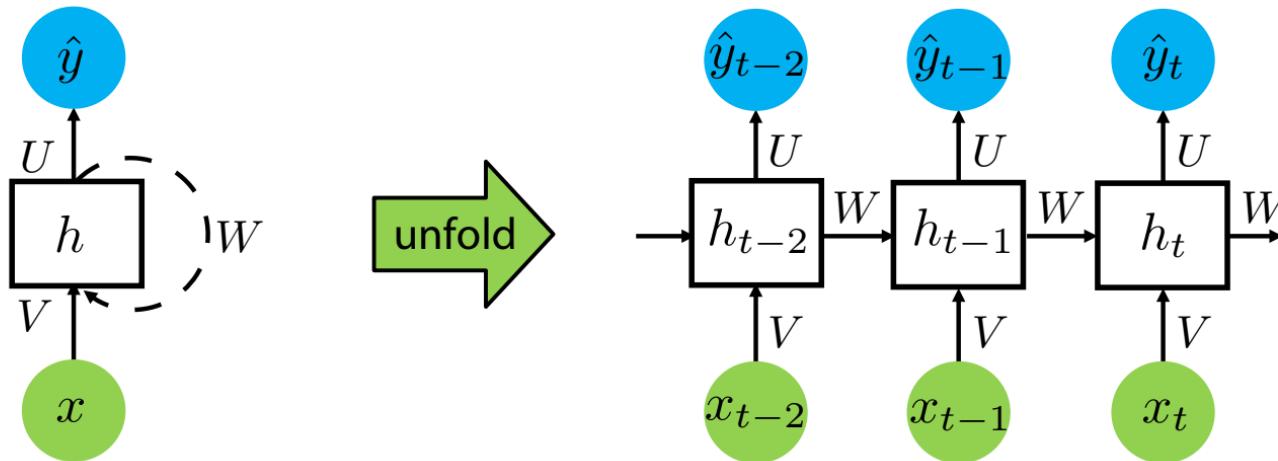
- Хорошее качество в самых разных прикладных задачах
- Маленькая размерность
- Близким словам соответствуют близкие вектора
- Есть механизм обработки новых слов на teste

Минусы  :

- Требуют большего корпуса чем count-based модели

Что дальше?

Рекуррентная нейронная сеть (RNN)



x - input

\hat{y} - output (prediction)

h - hidden state

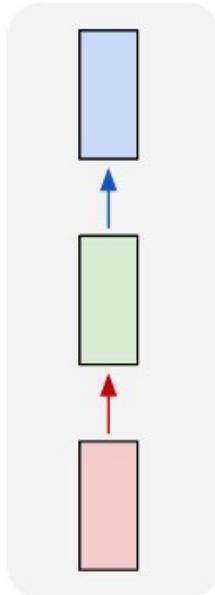
$$h_t = f_h(Vx_t + Wh_{t-1} + b_h)$$

$$\hat{y}_t = f_y(Uh_t + b_y)$$

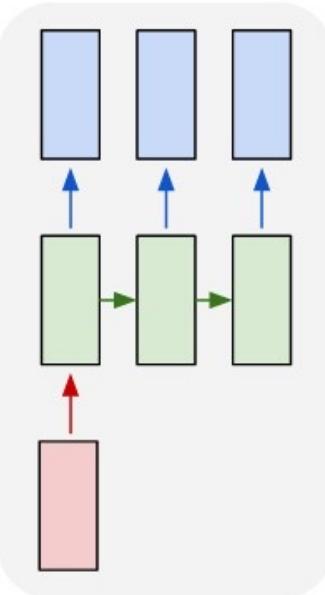
- Что будет эмбеддингами в RNN?
- Что будет токеном в RNN?

Возможности RNN

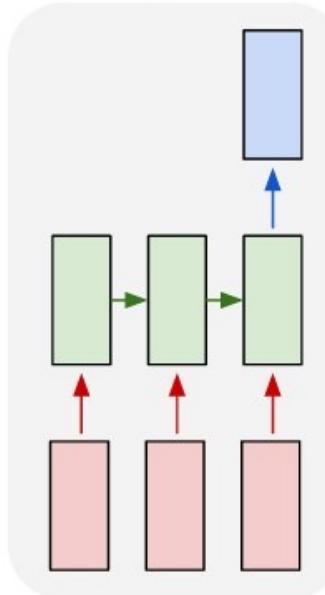
one to one



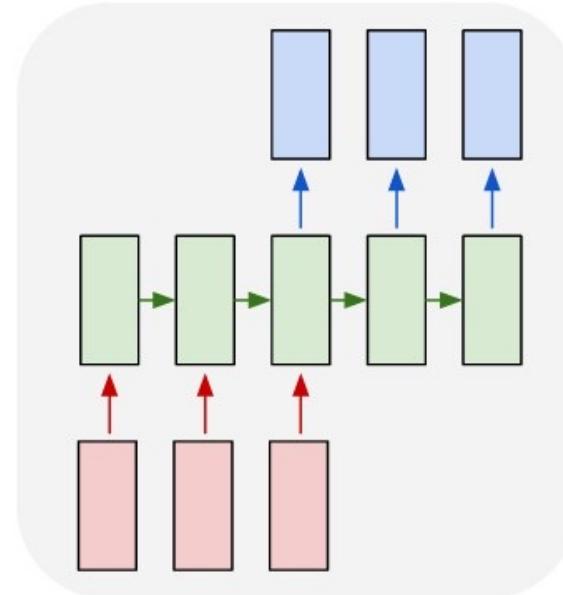
one to many



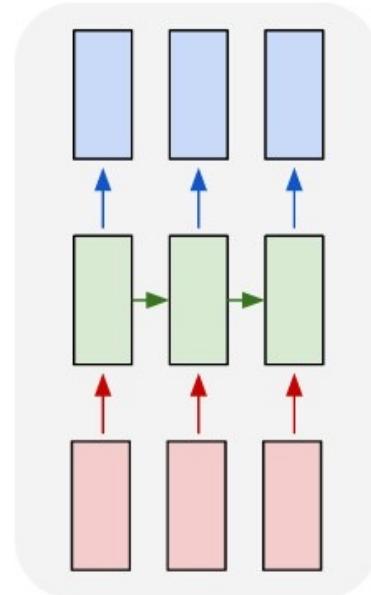
many to one



many to many



many to many



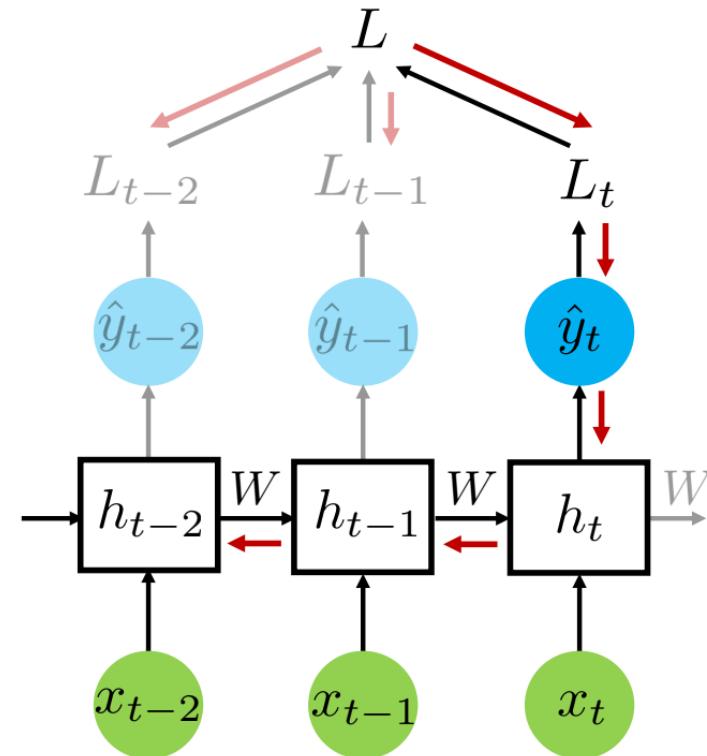
Backpropagation through time

$$\frac{\partial L}{\partial W} = \sum_{i=0}^T \frac{\partial L_i}{\partial W}$$

$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \frac{\partial h_t}{\partial W}$$

$$h_t = f_h(Vx_t + Wh_{t-1} + b_h)$$

This is **NOT** the only dependence!



$$\frac{\partial L_t}{\partial W} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \sum_{k=0}^t \left(\prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

Затухание и взрыв градиентов

$$\frac{\partial L_t}{\partial W} \propto \sum_{k=0}^t \left(\prod_{i=k+1}^t \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W}$$

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 < 1$$



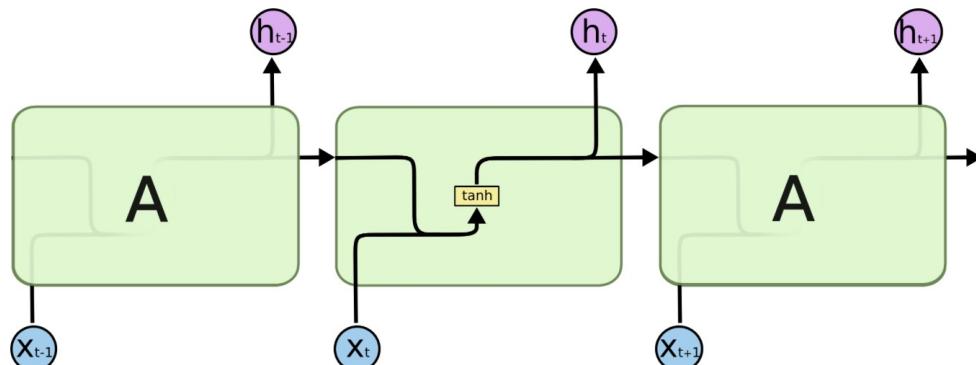
Vanishing gradients

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\|_2 > 1$$

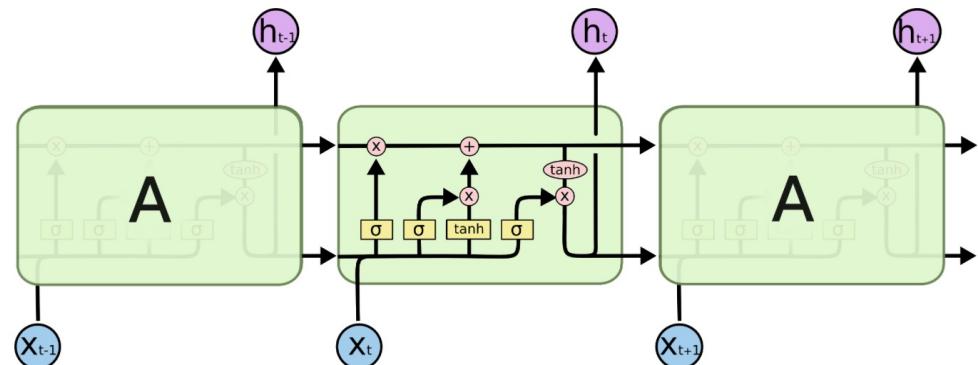


Exploding gradients

Long Short-Term Memory (LSTM)



The repeating module in a standard RNN contains a single layer.



The repeating module in an LSTM contains four interacting layers.

Могут быть bi-directional и многоуровневыми

Вопросы

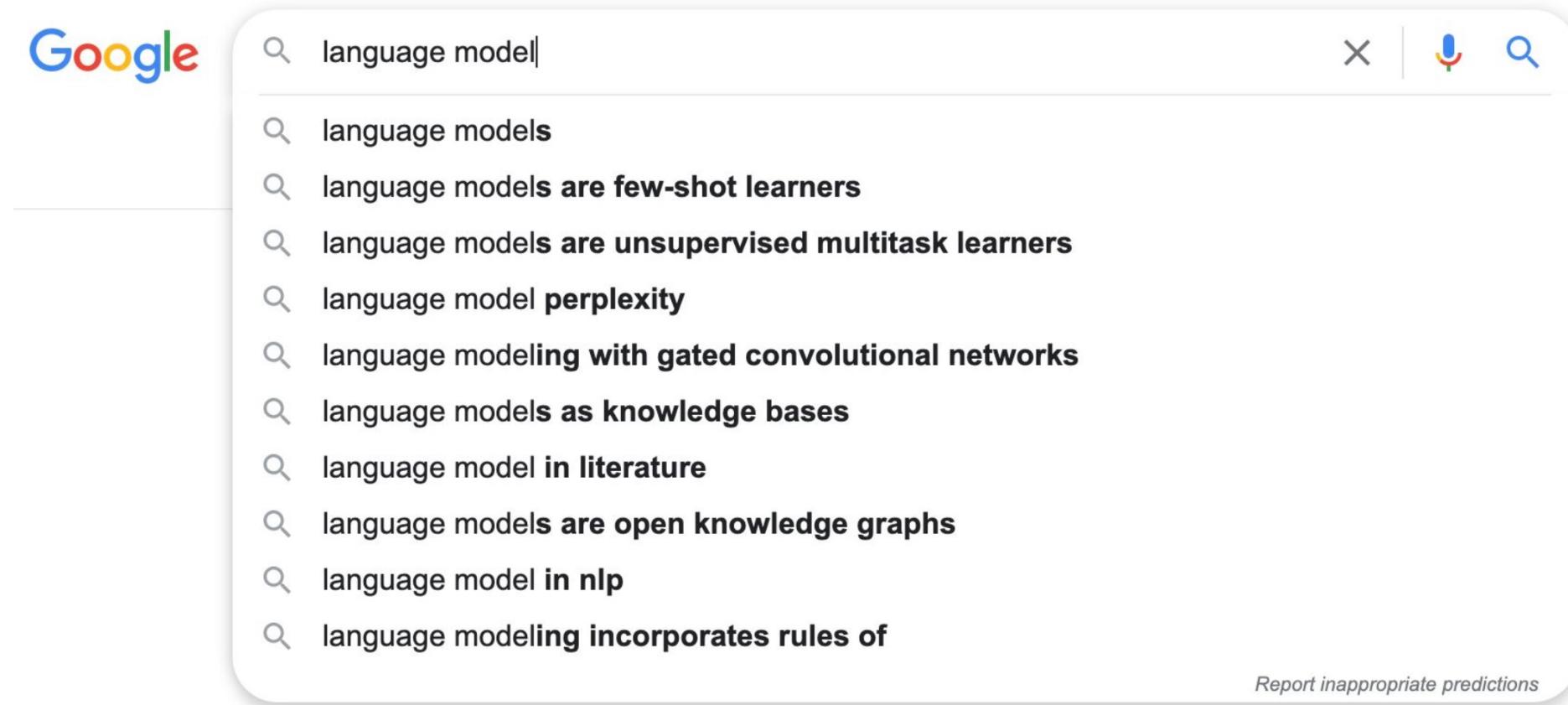
- Остались ли еще какие-то проблемы?
- Можем ли как-то использовать внутреннюю структуру языка для Self-Supervised обучения?

Языковое моделирование

Языковое моделирование

- Вероятностные:
 - count-based models
- Нейронные:
 - Recurrent Neural Networks, LSTM, GRU
- Пытаемся смоделировать язык

Применение



A screenshot of a Google search interface showing search suggestions for the query "language model". The suggestions are listed below the search bar, each preceded by a magnifying glass icon.

- language model
- language models
- language models **are few-shot learners**
- language models **are unsupervised multitask learners**
- language model **perplexity**
- language modeling **with gated convolutional networks**
- language models **as knowledge bases**
- language model **in literature**
- language models **are open knowledge graphs**
- language model **in nlp**
- language modeling **incorporates rules of**

At the bottom right of the suggestions list, there is a link: *Report inappropriate predictions*.

Постановка задачи

- Имеем:
 - $S = (w_1, w_2, \dots, w_n)$ – *sentence*, w_i – *word*, V – *vocabulary*
- Хотим:
 - $P(S) = P(w_1, w_2, \dots, w_n)$
- Есть ли сложности?

Постановка задачи

- Имеем:

$S = (w_1, w_2, \dots, w_n)$ – sentence, w_i – word, V – vocabulary

- Хотим:

$$P(S) = P(w_1, w_2, \dots, w_n)$$

- Раскладываем:

$$P(w_n | w_{n-1}, w_{n-2}, \dots, w_1)$$

Постановка задачи

- Chain Rule:

$$\begin{aligned} P(S) &= P(w_1, w_2, \dots, w_n) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1w_2) \dots P(w_n|w_1, \dots w_{n-1}) \\ &= \prod_{i=1}^n P(w_i|w < i) \end{aligned}$$

$$P(\text{«We study NLP»}) = P(\text{«We»}) \cdot P(\text{«study»}|\text{«We»}) \cdot \\ P(\text{«NLP»}|\text{«We study»})$$

Есть ли какие-то сложности?

Statistical LM vs Neural LM

Статистические модели

The independency assumption:

- “*The probability of a word depends on a fixed number of previous words*”

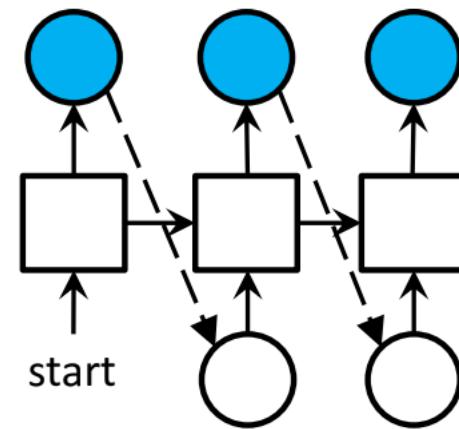
$$P(w_i | w_1, \dots, w_{i-1}) = P(w_i | w_{i-n+1}, \dots, w_{i-1}).$$

For example:

- $N=1$ (*unigram model*) – $P(w_i | w_1, \dots, w_{i-1}) = P(w_i)$
- $N=2$ (*bigram model*) – $P(w_i | w_1, \dots, w_{i-1}) = P(w_i | w_{i-1})$
- $N=3$ (*trigram model*) – $P(w_i | w_1, \dots, w_{i-1}) = P(w_i | w_{i-2}, w_{i-1})$

Генерация текста (инфереңс)

Next symbol/word



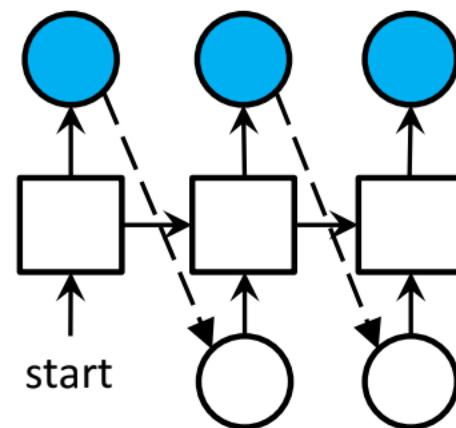
Current symbol/word

Как на обучении?

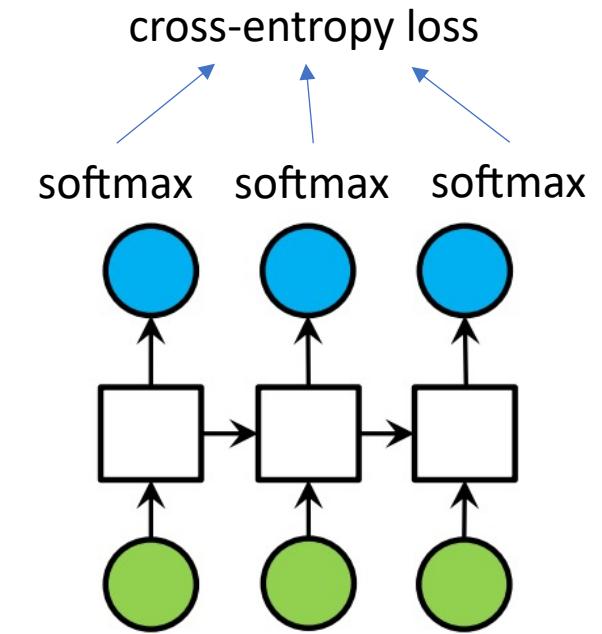
Обучение vs inference

Next symbol/word

Current symbol/word



Inference



Обучение

Стратегии декодирования

- Greedy
- Sampling
- Beam-Search
- Top-k / top-p
- ...

Машинный перевод

Машинный перевод

- Какая связь между языковым моделированием и машинным переводом?

Language Models: $P(y_1, y_2, \dots, y_n) = \prod_{t=1}^n p(y_t | y_{<t})$

Conditional

Language Models: $P(y_1, y_2, \dots, y_n, | \textcolor{green}{x}) = \prod_{t=1}^n p(y_t | y_{<t}, \textcolor{green}{x})$



condition on source x

Постановка задачи

Source sentence (x_1, x_2, \dots, x_n)

Target sentence (y_1, y_2, \dots, y_n)

Machine translation systems learn a function: $p(y|x, \theta)$

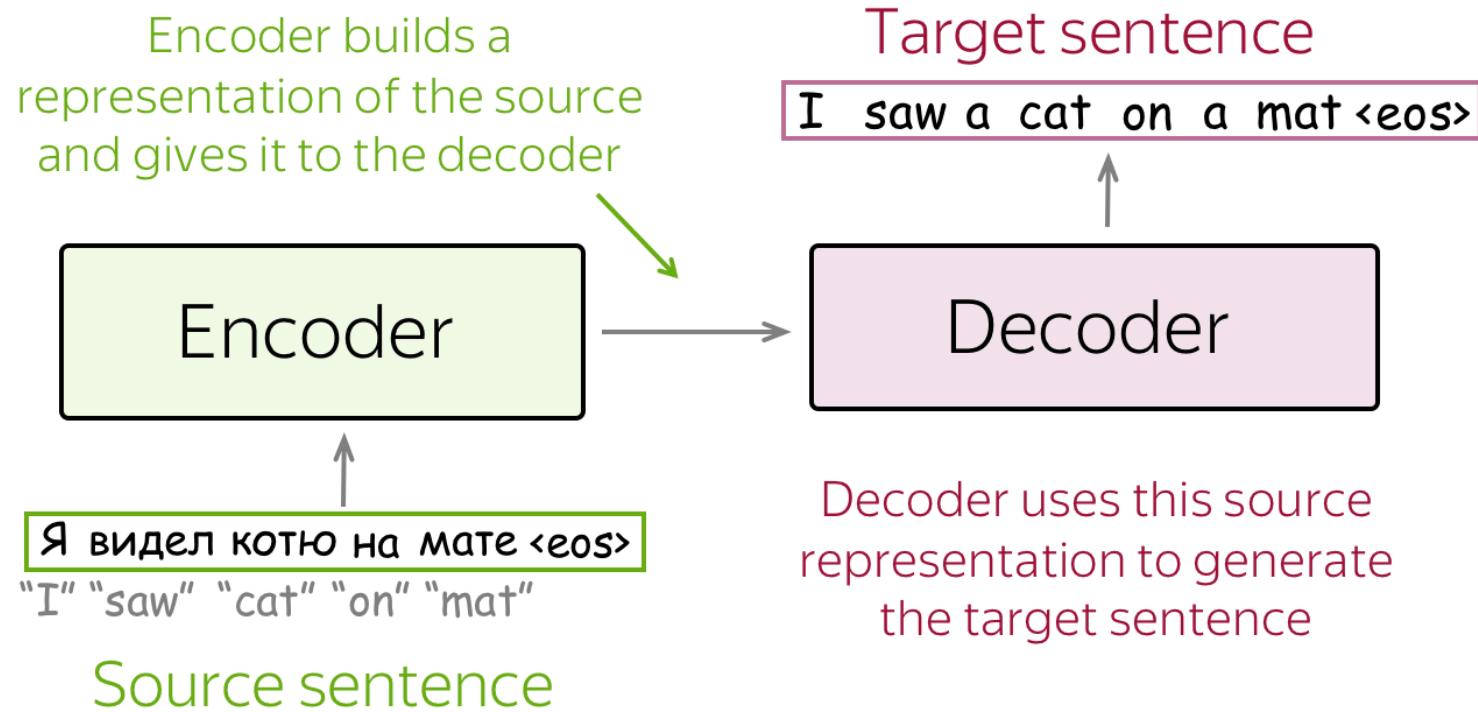
We try to find the target sequence that maximizes the

conditional probability: $y = \arg \max_y p(y|x, \theta)$

where θ – model parameters that determine probability distribution

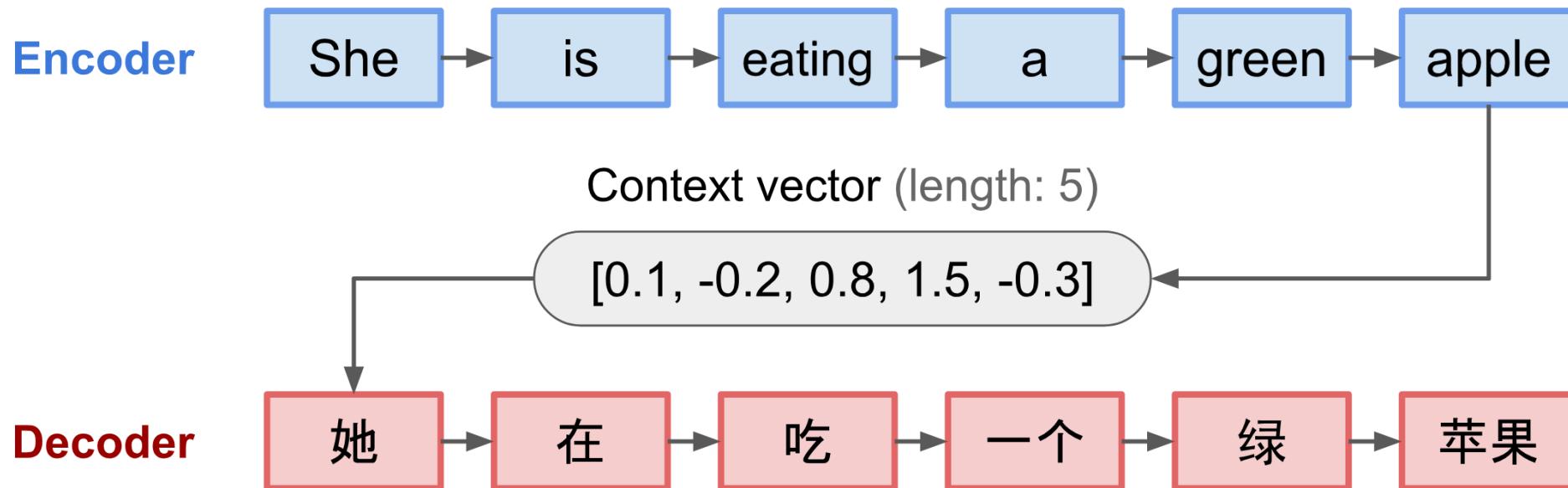
Постановка задачи

- Seq2seq
- Encoder - decoder

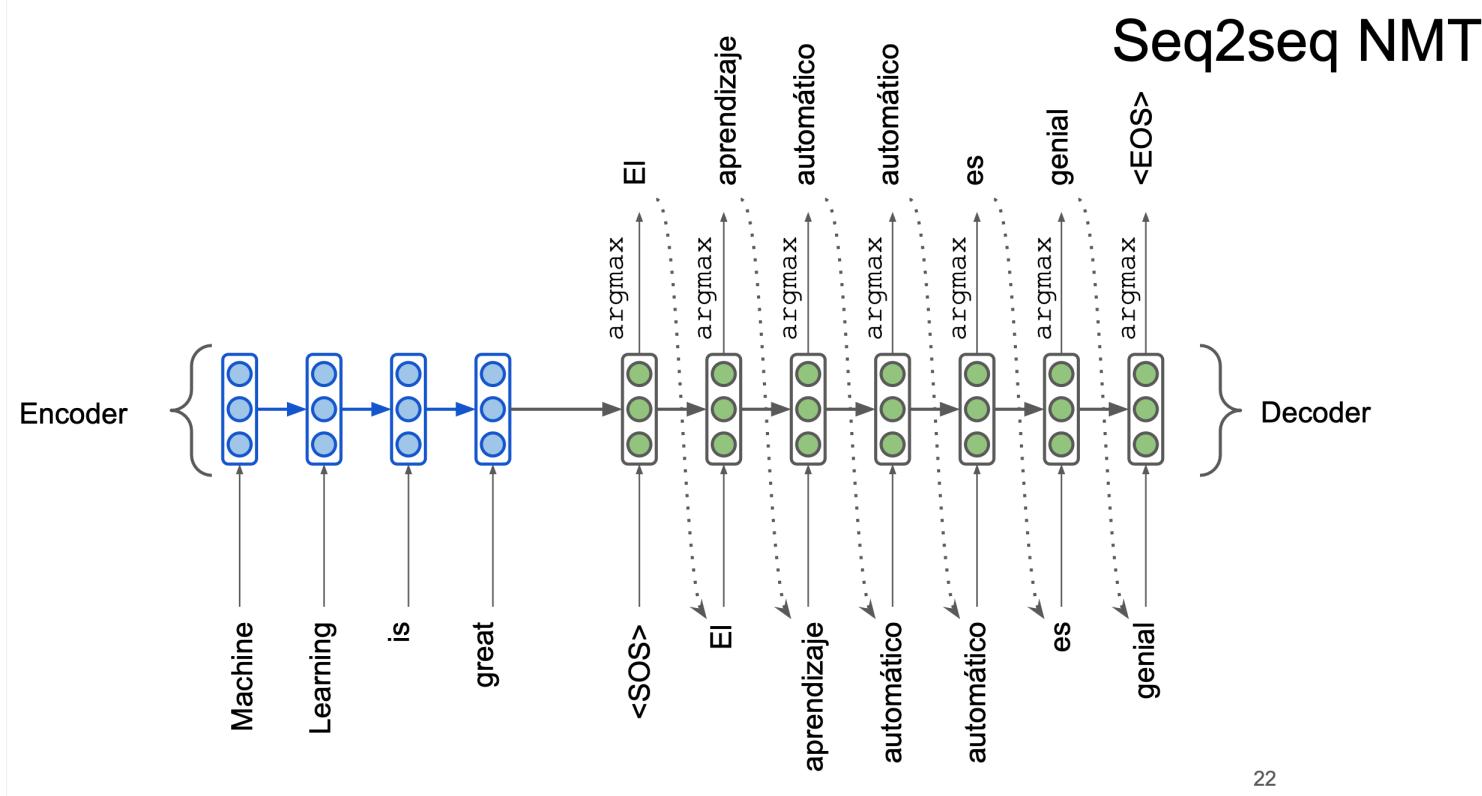


Постановка задачи

- Seq2seq
- Encoder - decoder



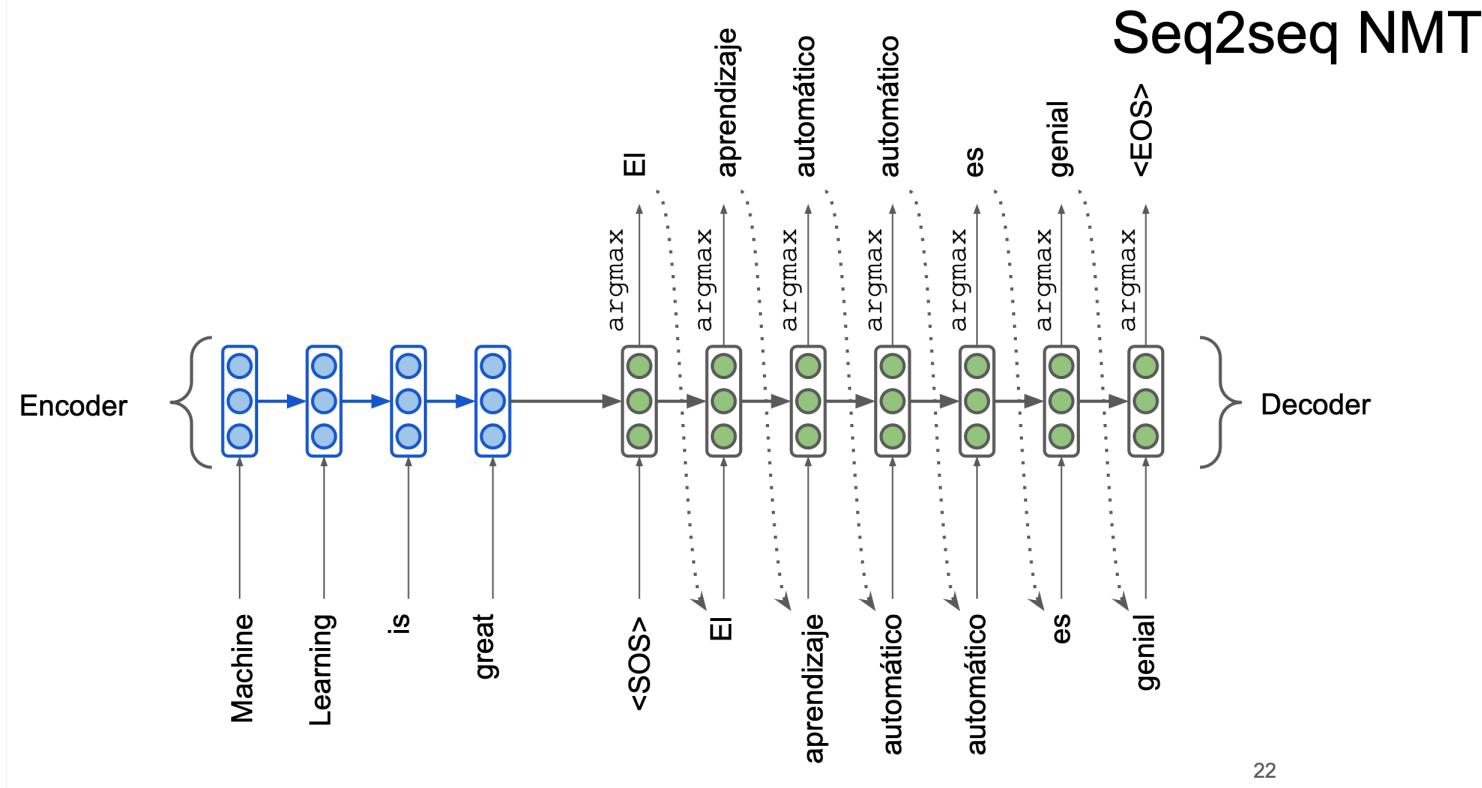
Машинный перевод



22

Это обучение или инференс?

Машинный перевод



22

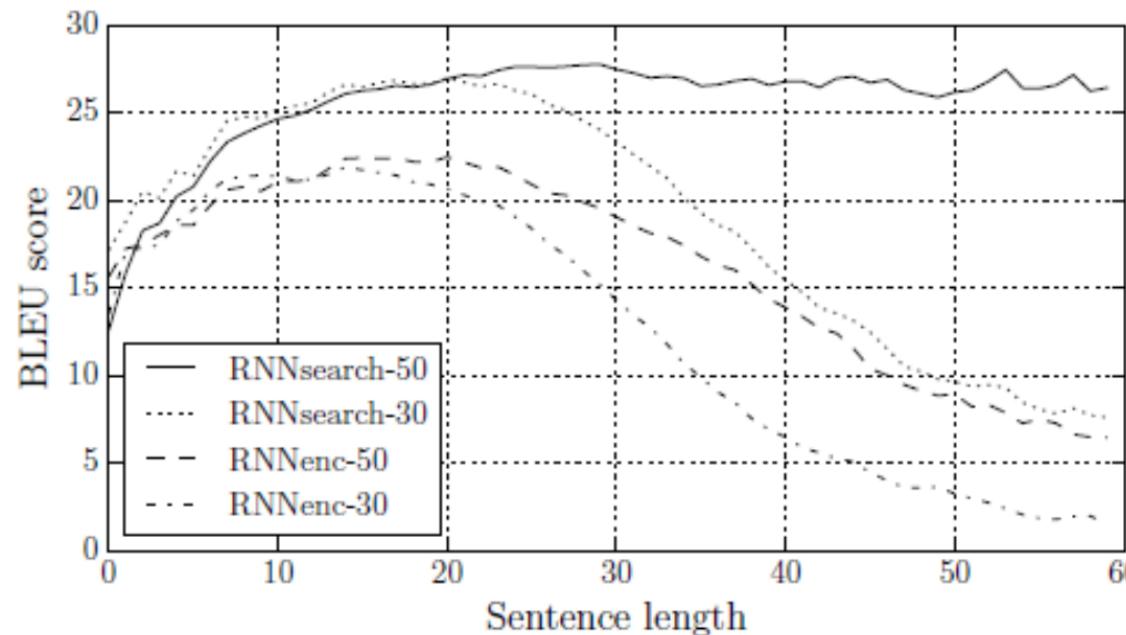
Это обучение или инференс?
Какая стратегия декодирования?

Вопросы

- Есть ли проблемы?

Вопросы

- Есть ли проблемы?

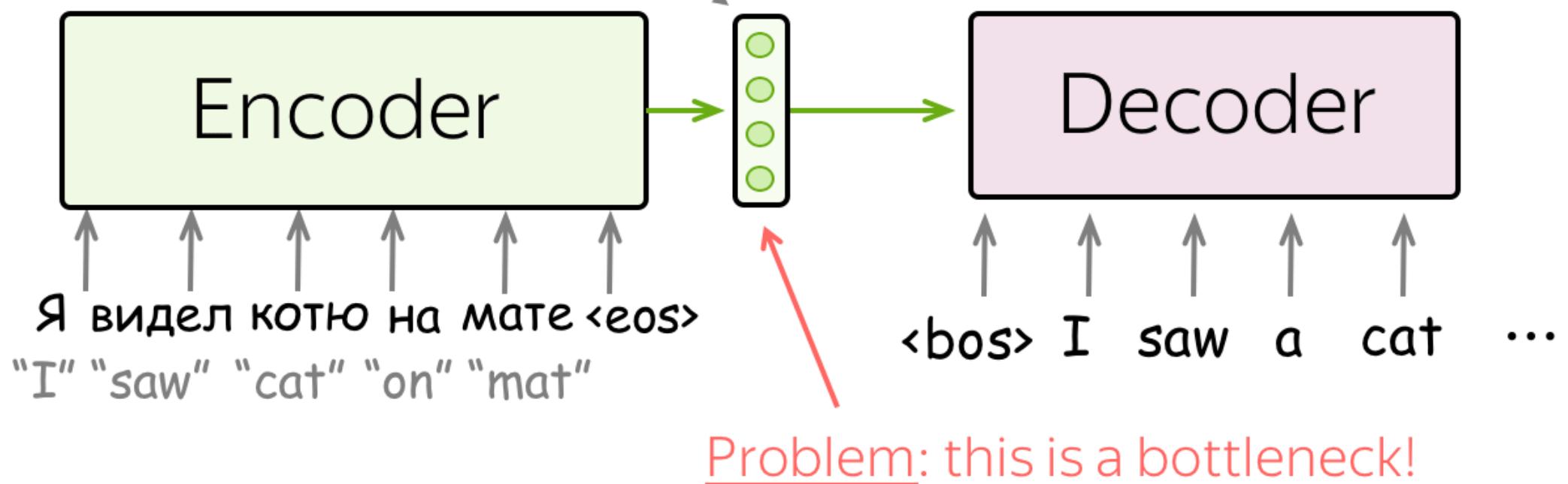


- Проблема и в LM и в NMT

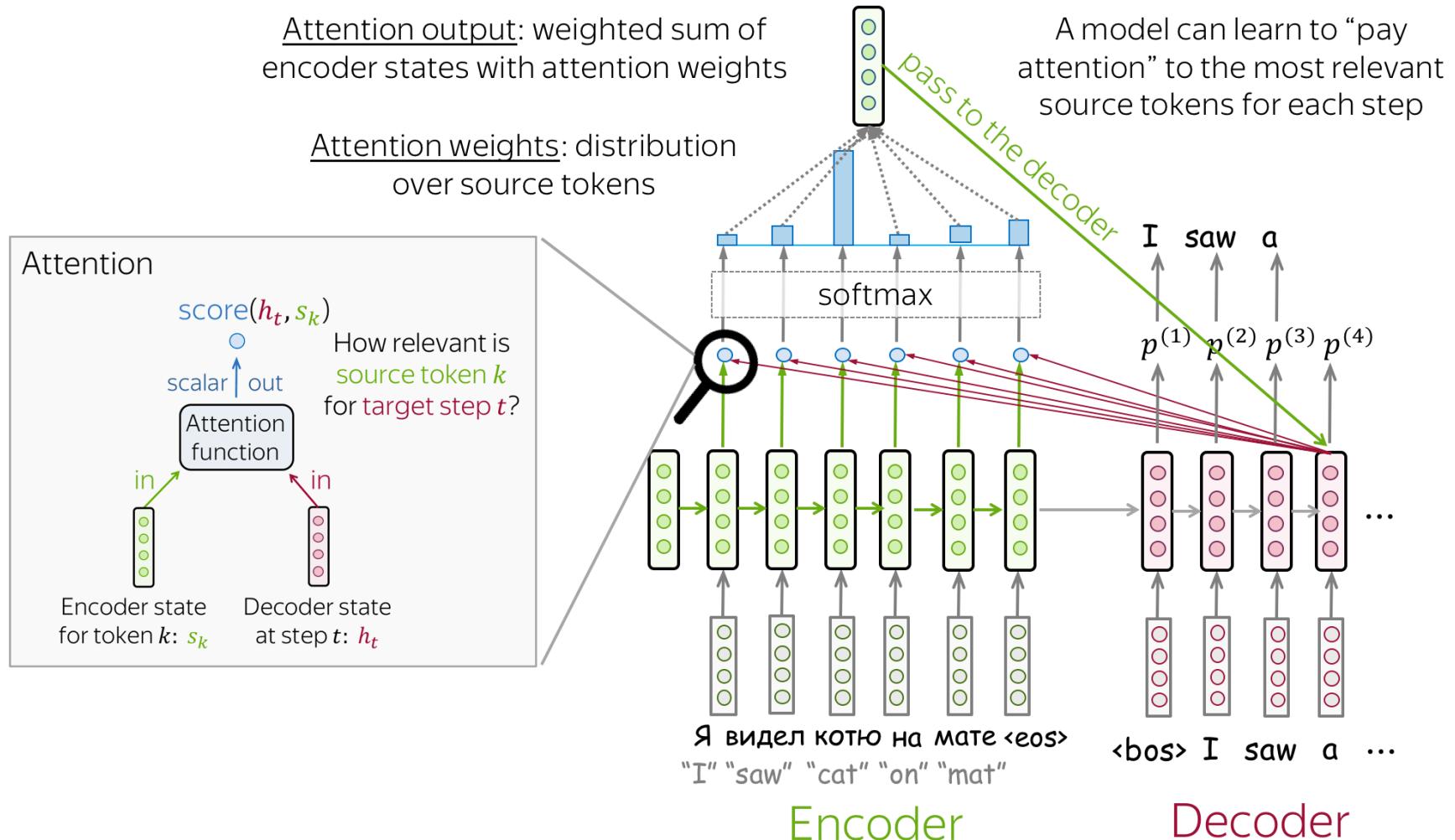
Attention (Born for translation)

Attention

We saw: encoder compresses the source into a single vector



Attention



TLDR

Даже если посмотреть на хронологию развития NLP, то тут все предельно логично и интуитивно:

- 1) Хотим работать с текстами - вот вам TF
- 2) Хотим в TF взвешивать слова пропорционально их разделяющей способности текстов - вот вам TF-IDF
- 3) Хотим заложить семантику в векторное представление (P.S. по факту TF-IDF - это тоже эмбеддинг)
- вот вам Word2Vec, Glove
- 4) Хотим побороть OOV проблему в Word2Vec - вот вам FastText
- 5) Хотим заложить контекстуализацию и избавиться от омонимии - вот вам ELMo
- 6) Что-то какой-то долгий этот ваш ELMo и RNN - вот вам Transformer и GPT
- 7) Хотим двунаправленность - вот вам BERT
- 8) Ну и так далее...