



# ТЕХНОСФЕРА

## Лекция 9 Нейронные сети для обработки естественного языка

Байгушев Данила

9 апреля 2021 г.

# Задачи NLP

- ▶ Машинный перевод
- ▶ Анализ тональности
- ▶ Чат-боты
- ▶ Понимание естественного языка
- ▶ Понимание изображений

# Entity tracking

mary got the milk there  
john moved to the bedroom  
sandra went back to the kitchen  
mary travelled to the hallway  
john got the football there  
john went to the hallway  
john put down the football  
mary went to the garden  
john went to the kitchen  
sandra travelled to the hallway  
daniel went to the hallway  
mary discarded the milk  
where is the milk ?

answer: garden

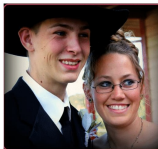
# Visual QA<sup>1</sup>

Who is wearing glasses?

man



woman

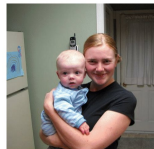


Where is the child sitting?

fridge



arms

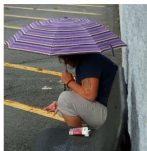


Is the umbrella upside down?

yes



no

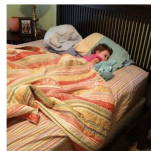


How many children are in the bed?

2



1



<sup>1</sup>Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering (CVPR 2017)

# Visual QA<sup>2</sup>

Answer: No



Answer: Yes



complementary scenes

Tuple: <girl, walking, bike>

Question: Is the girl walking the bike?

<sup>2</sup>Yin and Yang: Balancing and Answering Binary Visual Questions (CVPR 2016)

# Анализ тональности

## Отзыв положительный или отрицательный?

$y=1$  Мне очень понравился этот фильм. Никогда раньше ничего подобного не видел!!!)))

$y=0$  Ужасно! Ушла с середины фильма, тк больше невозможно было смотреть.

$y=0$  Ну да, конечно. Просто отличный фильм.

# Анализ тональности

## Отзыв положительный или отрицательный?

$y=1$  Мне очень понравился этот фильм. Никогда раньше ничего подобного не видел!!!)))

$y=0$  Ужасно! Ушла с середины фильма, тк больше невозможно было смотреть.

$y=0$  Ну да, конечно. Просто отличный фильм.

Простой подход: Bag-of-words + Logistic regression  
Какие есть проблемы у такого подхода?

# Анализ тональности

## Отзыв положительный или отрицательный?

$y=1$  Мне очень понравился этот фильм. Никогда раньше ничего подобного не видел!!!)))

$y=0$  Ужасно! Ушла с середины фильма, тк больше невозможно было смотреть.

$y=0$  Ну да, конечно. Просто отличный фильм.

Простой подход: Bag-of-words + Logistic regression Какие есть проблемы у такого подхода?

- ▶ Не учитывает сарказм
- ▶ Не учитывает схожесть слов (например, кот  $\leftrightarrow$  котенок)
- ▶ Не учитывает порядок слов



# Представление слов

## Задача

Сопоставить каждому слову  $w$  из словаря  $V$  вектор  $e(w)$ .

Подходы:

- ▶ One-hot encoding
- ▶ Counts
- ▶ CBOW
- ▶ Skip-grams

# One-hot encoding

Кодируем слово  $w_i$  вектором  $[0, 0, \dots, 0, \underbrace{1}_i, 0, \dots, 0]^T$  Плюсы:

- ▶ Просто реализовать
- ▶ Можно использовать разреженное представление

Минусы:

- ▶ Не учитывает близость слов
- ▶ Огромная размерность

# Counts

... and the cute **kitten** purred and then ...

... the cute **furry** cat purred and miaowed ...

... that small **kitten** miaowed and she ...

... the loud **furry** dog ran and bit ...

Словарь: bit, cute, furry, loud, miaowed, purred, ran, small

**kitten**: cute, purred, small, miaowed  $\Rightarrow [0, 1, 0, 0, 1, 1, 0, 1]^T$

**cat**: cute, furry, miaowed  $\Rightarrow [0, 1, 1, 0, 1, 0, 0, 0]^T$

**dog**: loud, furry, ran, bit  $\Rightarrow [1, 0, 1, 1, 0, 0, 1, 0]^T$

$$\text{sim}(w_1, w_2) = \frac{\langle w_1, w_2 \rangle}{\|w_1\| \cdot \|w_2\|}$$

# Embedding matrix

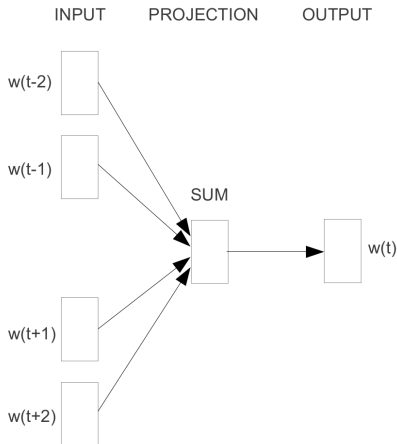
Матрица представлений:

$$E = \begin{bmatrix} \text{---}e_1\text{---} \\ \text{---}e_2\text{---} \\ \dots\dots\dots \\ \text{---}e_{|V|}\text{---} \end{bmatrix}$$

Каждая строка — представление одного слова.

Идея: обучим матрицу  $E$  при помощи нейронной сети.

# Continuous bag of words



Предсказываем пропущенное слово по контексту.

Представление слова:

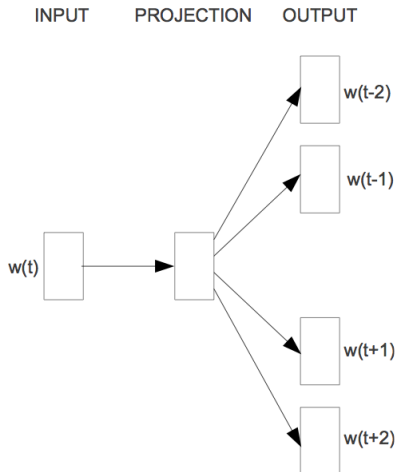
$$h_t = W \sum_{w \in \text{context}(w_t)} \text{one\_hot}(w)$$

$$P(w_i | \text{context}(w_i)) = \text{softmax}(W'h) [w_i]$$

Функция потерь:

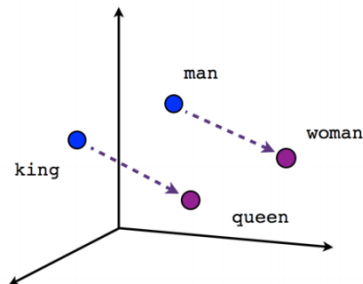
$$L = -\log P(w_i | \text{context}(w_i))$$

# Skip-gram

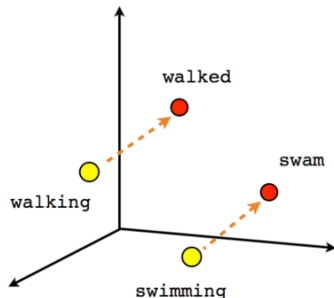


Предсказываем контекст по  
пропущенному слову

## word2vec: арифметика



Male-Female



Verb tense

$$e(\langle \text{king} \rangle) - e(\langle \text{man} \rangle) + e(\langle \text{woman} \rangle) \simeq e(\langle \text{queen} \rangle)$$

$$e(\langle \text{swimming} \rangle) + e(\langle \text{walked} \rangle) - e(\langle \text{walking} \rangle) \simeq e(\langle \text{swam} \rangle)$$

# Проблема Softmax

- ▶ Linear + softmax:  $P(t|x) = \text{Softmax}(Wx) = \frac{e^{\langle x, w_t \rangle}}{\sum_j e^{\langle x, w_j \rangle}}$
- ▶ Forward pass:  $\mathcal{O}(|V|d^2)$
- ▶ Backward pass:  $\mathcal{O}(|V|d)$
- ▶ Prediction:  $\mathcal{O}(|V|d^2)$
- ▶ Для больших словарей (сотни тысяч) работает очень долго

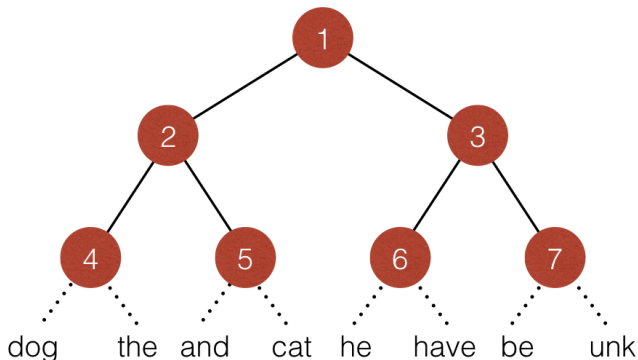
Решения:

- ▶ Sampled Softmax / Noise-contrastive estimation
- ▶ Иерархический Softmax
- ▶ Дифференцированный Softmax



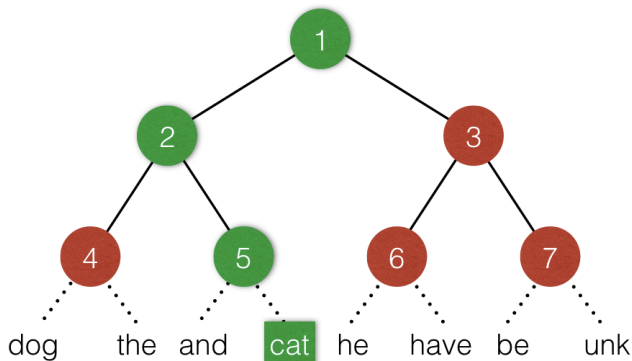
# Иерархический Softmax

$$P(\text{«cat»}|\text{context}) = P(1 \rightarrow 2|\text{context}) \times \\ P(2 \rightarrow 5|\text{context}) \times \\ P(5 \rightarrow \text{«cat»}|\text{context})$$



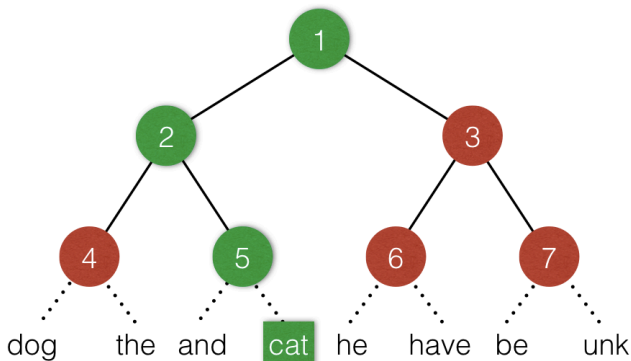
# Иерархический Softmax

$$P(\text{«cat»}|\text{context}) = P(1 \rightarrow 2|\text{context}) \times \\ P(2 \rightarrow 5|\text{context}) \times \\ P(5 \rightarrow \text{«cat»}|\text{context})$$



# Иерархический Softmax

$$P(\text{«cat»}|x) = (1 - \sigma(b_1 + h_1^T x)) \times \\ \sigma(b_2 + h_2^T x) \times \\ \sigma(b_5 + h_5^T x)$$



# Иерархический Softmax

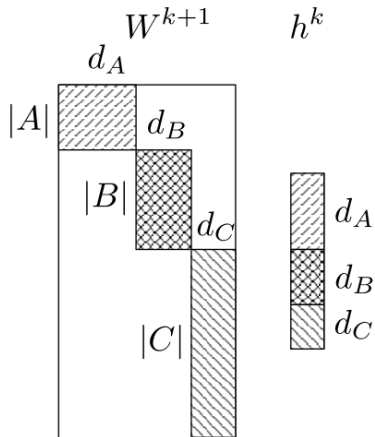
Как построить дерево?

- ▶ Случайно приписать слова листьям
- ▶ Иерархическая кластеризация представлений слов
- ▶ Код Хаффмана

В train-time сложность  $O(\log_2 N)$

В test-time требуется вычислять значения во всех листьях  $\Rightarrow$  медленнее чем обычный softmax (хуже параллелится).

# Дифференцированный Softmax



Редким словам сопоставляем короткие вектора, частым — длинные.  
Скорость в train-time и test-time одинаковая.

Чат-боты

# Известные боты

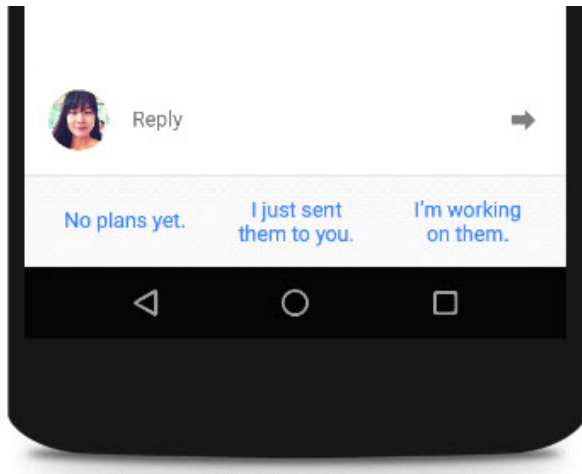


Figure: Google smart reply

# Известные боты



Figure: Microsoft Tay, твиттер бот



# Известные боты



В ответ 2 days til election



**DeepDrumpf** @DeepDrumpf · 31 окт.

I've got to win first. That's what I do. You have, right here in Colorado and a lot of the states, voting for ISIS on Nov 8th. [@Donna\\_West](#)



34



90



В ответ Jon Favreau



**DeepDrumpf** @DeepDrumpf · 26 окт.

No, Abraham Latino is poisoning our country. Impossible to sell our product. There will be no amnesty, but this is locker room talk. [@jonfavs](#)



57



127



Figure: DeepDrumpf, твиттер бот

# Известные боты

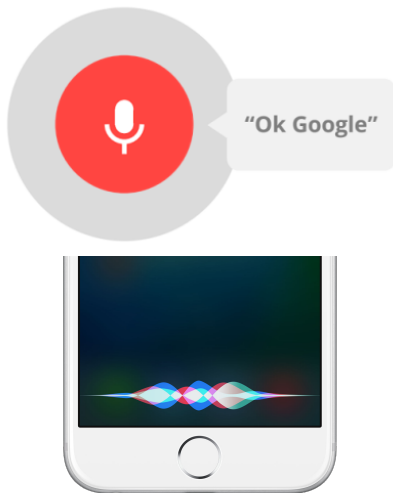
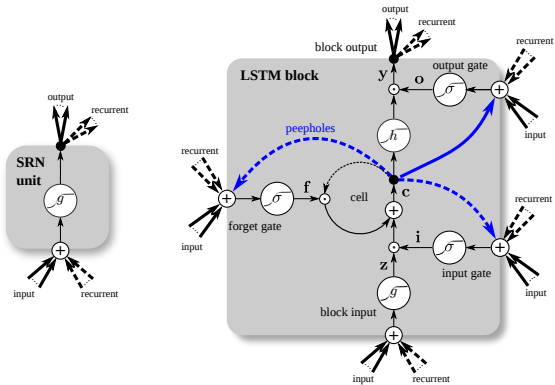











Figure: Голосовые помощники

## LSTM сети



### Legend

-  unweighted connection
-  weighted connection
-  connection with time-lag
-  branching point
-  multiplication
-  sum over all inputs
-  gate activation function (always sigmoid)
-  input activation function (usually tanh)
-  output activation function (usually tanh)

# Conversation vs Goal

## Общение для общения

- ▶ Ответы должны быть более-менее релевантными
- ▶ Надо поддерживать контекст беседы
- ▶ Ответы должны быть разнообразными
- ▶ Метрика: A/B тесты, ассесоры

## Общение для достижения цели

- ▶ Реплики бота должны приближать диалог к цели
- ▶ Надо поддерживать контекст беседы
- ▶ Метрика: Accuracy, Precision, Recall, ...

# Вероятностная постановка

По последовательности слов  $w_1, w_2, \dots, w_n$  надо найти распределение  $P(w_1, w_2, \dots, w_n)$ .

Chain rule:  $P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, w_2, \dots, w_{i-1})$

Требуется научиться генерировать следующее слово по предыдущим.

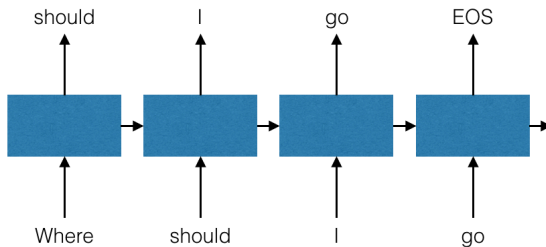
# N-grams

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1}) \simeq \prod_{i=1}^n P(w_i | w_{i-k}, \dots, w_{i-1})$$

Обучение: считаем количество вхождений  $w_{i-k}, \dots, w_{i-1}, w_i$  и нормируем, чтобы получить вероятности.

- ▶ Чем больше  $k$ , тем более общая/переобученная модель
- ▶ Требуется много памяти
- ▶ Ограниченная длина контекста

# Нейронные сети



$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$

$$P(w_i | w_1, \dots, w_{i-1}, c_i) \simeq f(w_i | c_{i-1})$$

Где  $f$  — нейронная сеть

# Задача генерации ответа

По последовательности слов  $q_i$  надо найти распределение на последовательность  $a_i$ :  $P(a|q) = ?$ .



# Seq2Seq

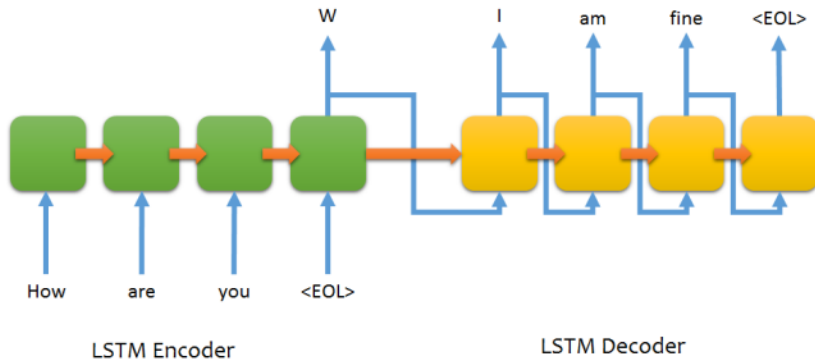
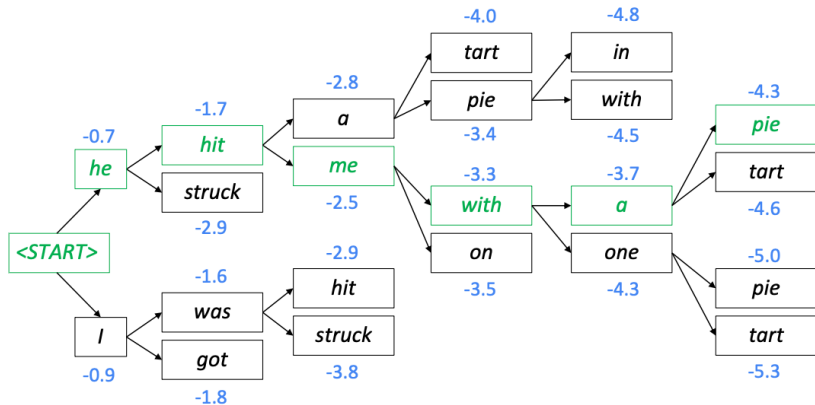


Figure: Перевод последовательностей друг в друга

# Teacher forcing in beam search

## Beam search decoding: example

Beam size =  $k = 2$ . Blue numbers =  $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$



# Проблемы

## Генерация

Умеем вычислять  $P(a|q)$  для всех возможных ответов.

Предсказание:  $a_{MP} = \arg \max_a P(a|q)$  или  $a_S \sim P(a|q)$ .

Для  $a_{MP}$  можно использовать beam search.

## Разнообразие

После beam search часто получаются частотные ответы: «да», «нет», «не знаю».

Можно обучить две сети:  $P(a|q)$  и  $P(q|a)$ .

$$a_{MP} = \arg \max_a \left[ \lambda P(a|q) + (1 - \lambda) P(q|a) \right]$$

# За рамками лекции

- ▶ Поддержание диалога: HRED
- ▶ Работа со словами не из словаря
- ▶ RL для NMT, DualNMT
- ▶ Современные модели (следующая лекция)

# Вопросы

