

Глубокое обучение

Дмитрий Никулин

16 июня 2021 г.

Неделя 12: Из слов в вектора и обратно

Agenda

- ML — это максимизация правдоподобия
- Небольшое введение в анализ текстов
- Представления для текстов, эмбеддинги

ML — это максимизация
правдоподобия

$$\mathcal{L}(\theta) = \mathbb{P}_\theta(x) \rightarrow \max_{\theta}$$

Функция правдоподобия (likelihood function)

- Пусть даны:
 - Параметрическое семейство вероятностных распределений $\{\mathbb{P}_\theta\}_{\theta \in \Theta}$;
 - Выборка (независимых одинаково распределённых) $x_1, \dots, x_n \sim \mathbb{P}_{\theta^*}$ для некоторого θ^* .
- Тогда функция

$$\mathcal{L}(\theta) = \mathbb{P}_\theta(x_1, \dots, x_n) = \prod_{i=1}^n \mathbb{P}_\theta(x_i)$$

называется функцией правдоподобия (likelihood function).

- Эта функция по параметрам θ определяет, насколько хорошо модель \mathbb{P}_θ с этими параметрами описывает датасет.

Функция правдоподобия (likelihood function)

- Обычно мы хотим максимизировать $\mathcal{L}(\theta)$ по θ , то есть найти параметры, лучше всего описывающие датасет.
- Часто вместо $\mathcal{L}(\theta)$ удобнее рассматривать её логарифм, тогда произведение превращается в сумму:

$$\log \mathcal{L}(\theta) = \log \mathbb{P}_\theta(x_1, \dots, x_n) = \sum_{i=1}^n \log \mathbb{P}_\theta(x_i) \rightarrow \max_\theta$$

Пример (задача регрессии)

- $\theta \in \mathbb{R}^d$ — коэффициенты линейной регрессии
- Элемент датасета — это пара (x_i, y_i) ($x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$)
- Вероятностная модель (σ может быть любым):

$$\mathbb{P}_\theta(x_i, y_i) = \mathcal{N}(\theta \cdot x_i, \sigma^2)(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\theta \cdot x_i - y_i)^2}{2\sigma^2}\right)$$

- Функция правдоподобия:

$$\mathcal{L}(\theta) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\theta \cdot x_i - y_i)^2}{2\sigma^2}\right)$$

Пример (задача регрессии)

- Логарифм правдоподобия:

$$\begin{aligned}\log \mathcal{L}(\theta) &= \sum_{i=1}^n \left[\log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{(\theta \cdot x_i - y_i)^2}{2\sigma^2} \right] \\ &= n \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2} \sum_{i=1}^n (\theta \cdot x_i - y_i)^2 \rightarrow \max_{\theta}\end{aligned}$$

- Логарифм правдоподобия достигает максимума, когда

$$\sum_{i=1}^n (\theta \cdot x_i - y_i)^2 \rightarrow \min_{\theta}$$

- Мы только что вывели MSELOSS.

Пример (задача классификации на 2 класса)

- $\theta \in \mathbb{R}^d$ — коэффициенты логистической регрессии
- Элемент датасета — это пара (x_i, y_i) ($x_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$)
- Вероятностная модель:

$$\begin{aligned}\mathbb{P}_\theta(x_i, y_i) = \text{Bernoulli}(\sigma(\theta \cdot x_i))(y_i) &= \begin{cases} 1 - \sigma(\theta \cdot x_i), & y_i = 0 \\ \sigma(\theta \cdot x_i), & y_i = 1 \end{cases} \\ &= (1 - \sigma(\theta \cdot x_i))^{1-y_i} \cdot (\sigma(\theta \cdot x_i))^{y_i}\end{aligned}$$

- Функция правдоподобия:

$$\mathcal{L}(\theta) = \prod_{i=1}^n \left[(1 - \sigma(\theta \cdot x_i))^{1-y_i} \cdot (\sigma(\theta \cdot x_i))^{y_i} \right]$$

Пример (задача классификации на 2 класса)

- Логарифм правдоподобия:

$$\log \mathcal{L}(\theta) = \sum_{i=1}^n [(1 - y_i) \log (1 - \sigma(\theta \cdot x_i)) + y_i \log (\sigma(\theta \cdot x_i))] \rightarrow \max_{\theta}$$

- Логарифм правдоподобия достигает максимума, когда

$$-\sum_{i=1}^n [(1 - y_i) \log (1 - \sigma(\theta \cdot x_i)) + y_i \log (\sigma(\theta \cdot x_i))] \rightarrow \min_{\theta}$$

- Мы только что вывели бинарную кросс-энтропию (`BCEWithLogitsLoss` в PyTorch).

Пример (задача классификации на C классов)

- $\theta \in \mathbb{R}^{C \times d}$ — коэффициенты мультиклассовой логистической регрессии
- Элемент датасета — это пара (x_i, y_i) ($x_i \in \mathbb{R}^d$, $y_i \in \{1, \dots, C\}$)
- Вероятностная модель:

$$\mathbb{P}_\theta(x_i, y_i) = \text{Categorical}(\text{Softmax}(\theta_1 \cdot x_i, \dots, \theta_C \cdot x_i))(y_i) =$$

$$\text{Softmax}_{y_i}(\theta_1 \cdot x_i, \dots, \theta_C \cdot x_i) = \frac{\exp(\theta_{y_i} \cdot x_i)}{\sum_{c=1}^C \exp(\theta_c \cdot x_i)}$$

- Функция правдоподобия:

$$\mathcal{L}(\theta) = \prod_{i=1}^n [\text{Softmax}_{y_i}(\theta_1 \cdot x_i, \dots, \theta_C \cdot x_i)]$$

Пример (задача классификации на C классов)

- Логарифм правдоподобия:

$$\log \mathcal{L}(\theta) = \sum_{i=1}^n \left[\text{LogSoftmax}_{y_i} (\theta_1 \cdot x_i, \dots, \theta_C \cdot x_i) \right] \rightarrow \max_{\theta}$$

- Логарифм правдоподобия достигает максимума, когда

$$-\sum_{i=1}^n \left[\text{LogSoftmax}_{y_i} (\theta_1 \cdot x_i, \dots, \theta_C \cdot x_i) \right] \rightarrow \min_{\theta}$$

- Мы только что вывели CrossEntropyLoss.

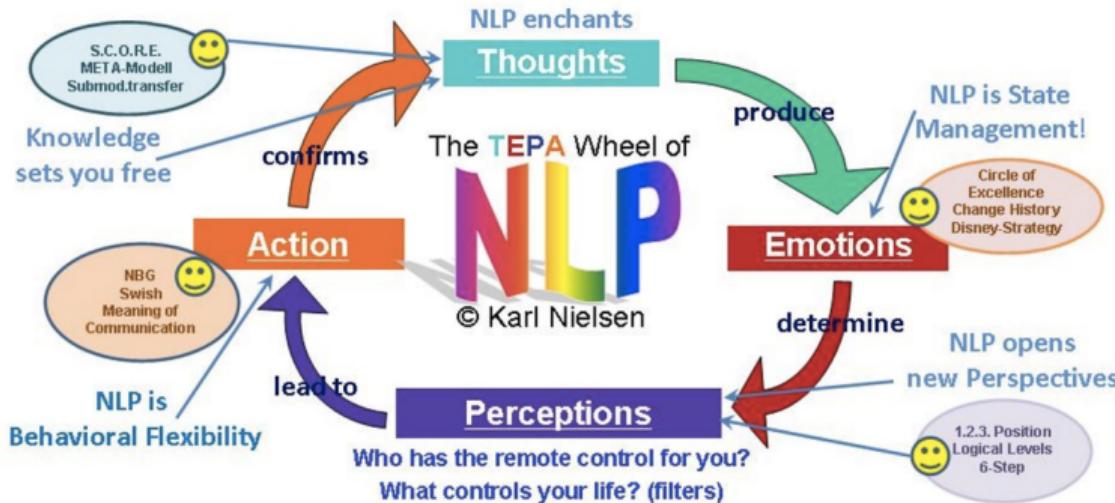
Резюме

- Стандартные лоссы (`MSELoss`, `BCEWithLogitsLoss`, `CrossEntropyLoss`) появляются при максимизации логарифма правдоподобия в понятных вероятностных моделях.
- Сегодня мы опишем некоторую вероятностную модель для текста, максимизация логарифма правдоподобия для которой даст нам `word2vec`.
- Почти всё машинное обучение — это максимизация правдоподобия.

NLP (Natural Language Processing)



NLP, the freedom in Thinking, Feeling, Perceiving and Behavior





freedom in thinking, receiving, behavior

Feelings

S.C.O.R.E.
META-Modell
Submod.transfer



Knowledge
sets you free

confirms

NBG
Swish
Meaning of
Communication

Action

NLP is
Behavioral F

Perceptions

Who has the remote control for you?
What controls your life? (filters)

produce

Emotions

NLP is State
Management!

Circle of
Excellence
Change History
Disney-Strategy

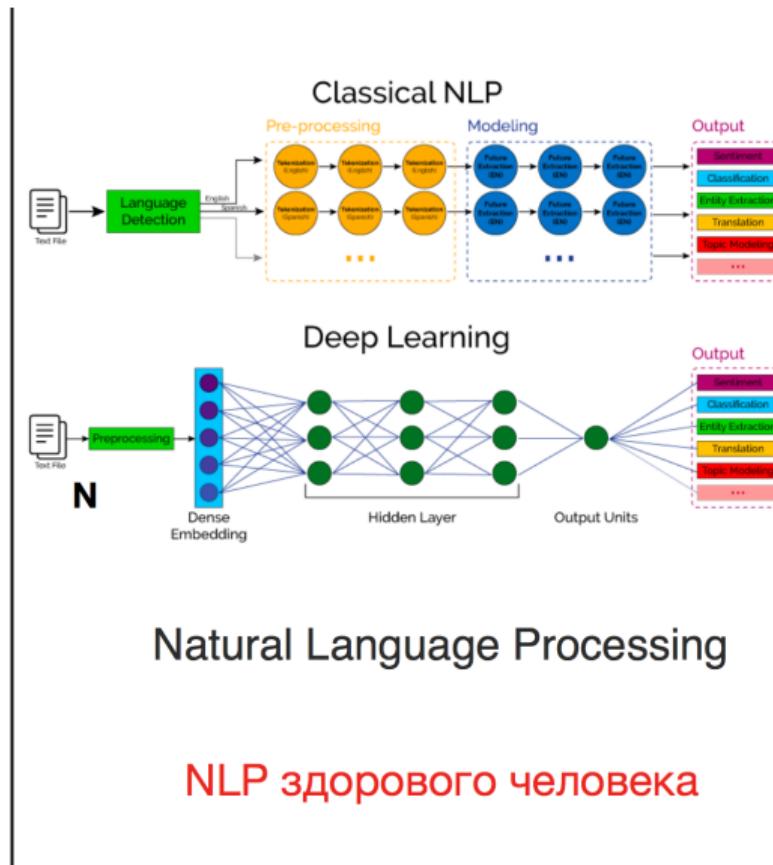
NLP opens
new Perspectives

1,2,3, Position
Metaphorical Levels
One-Step

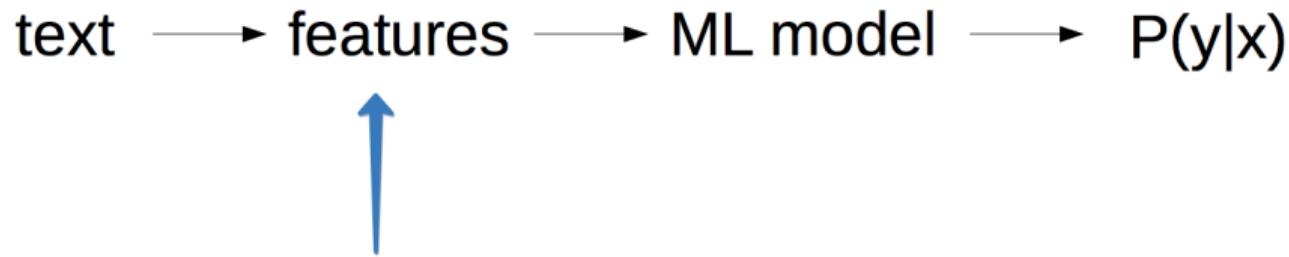


Neuro-linguistic programming

NLP курильщика

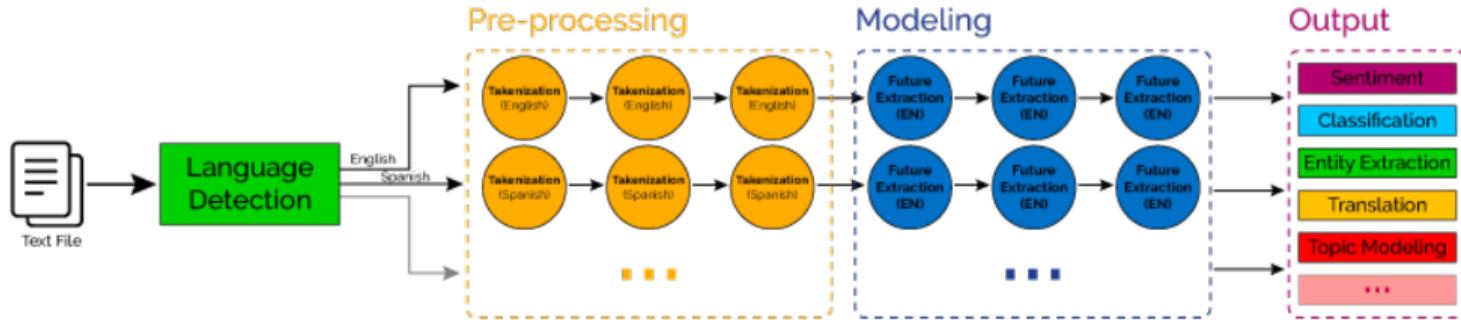


Модели на текстах

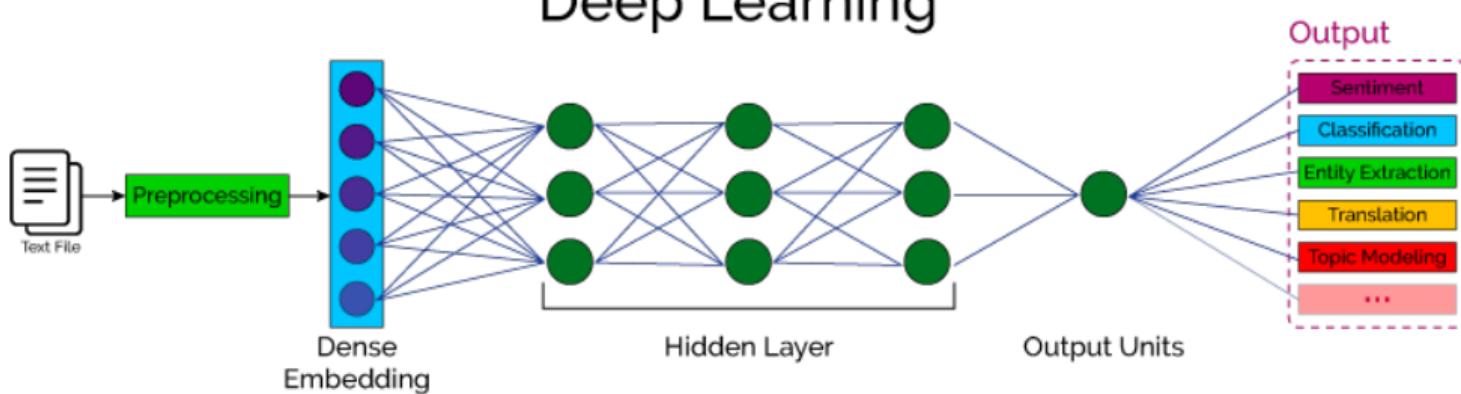


Как представить текст
в виде, который могла
бы понять модель?

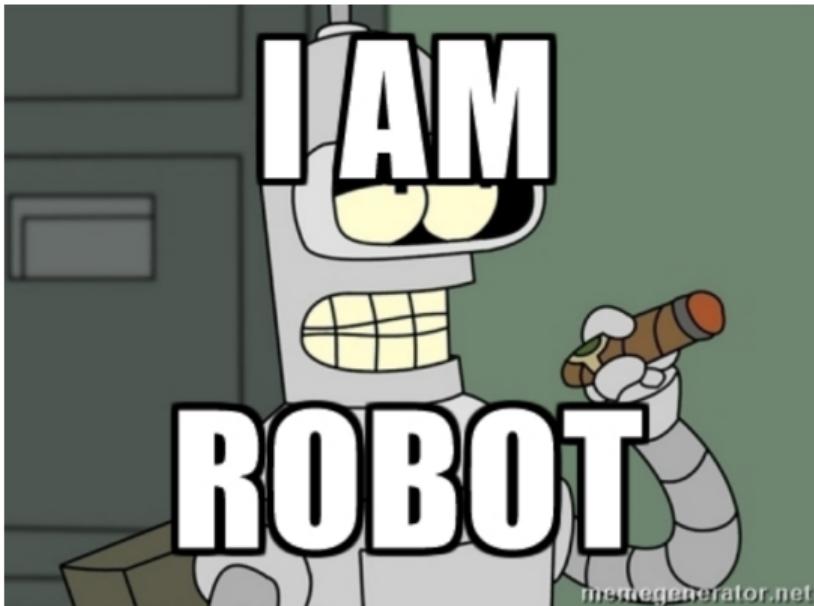
Classical NLP



Deep Learning



Что такое текст?



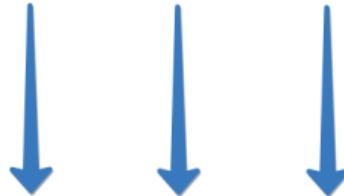
- Текст (документ) — это последовательность токенов (слов)
- Токен (слово) — это последовательность символов

Гипотеза мешка слов

- **Гипотеза мешка слов:** нам плевать на взаимное расположение слов. Порядок слов в предложении никак не сказывается на его смысле.
Следуя гипотезе, мы теряем часть информации.
- Рассматриваем каждое слово, как переменную, у которой столько же значений, сколько слов в словаре \Rightarrow большое пространство признаков.

One-Hot Encoding

1. Нежился на пляже
2. Копали яму на пляже
3. Копал картошку
4. Ел картошки и картошку



	нежиться	пляж	копать	яма	картошка	есть
1	1	1	0	0	0	0
2	0	1	1	1	0	0
3	0	0	1	0	1	0
4	0	0	0	0	2	1

One-Hot Encoding

- В словаре много слов, у нас будет слишком много признаков
- В векторе нет никакой информации о смысле слова
- Семантически похожие тексты могут иметь очень разные представления
- Непонятно, что делать, если в тексте появляется какое-то новое слово

Классический анализ текстов в одном слайде



Порядок слов неважен

Неважен слов порядок

Слов порядок неважен

Он был хорошим человеком и джедаем.
Люди держат деньги в банке.
Падаван, дай мне огурец из банки.



1. токенизация
2. очистка от стоп-слов

[~~он~~, был, хорошим, человеком, ~~и~~, джедаем]
[люди, держат, деньги, ~~в~~, банке]
[падаван, дай, ~~мне~~, огурец, ~~из~~, банки]

лемматизация

[быть, хороший, человек, джедай]
[человек, держать, деньги, банк]
[падаван, дать, огурец, банка]

3. нормализация

стемминг

[бы, хорош, чел, джед]
[люд, держ, ден, банк]
[падаван, дай, огур, банк]

5. ОНЕ или tf-idf,
а затем
моделирование

4. очистка от слишком редких слов

Из слов в вектора и обратно



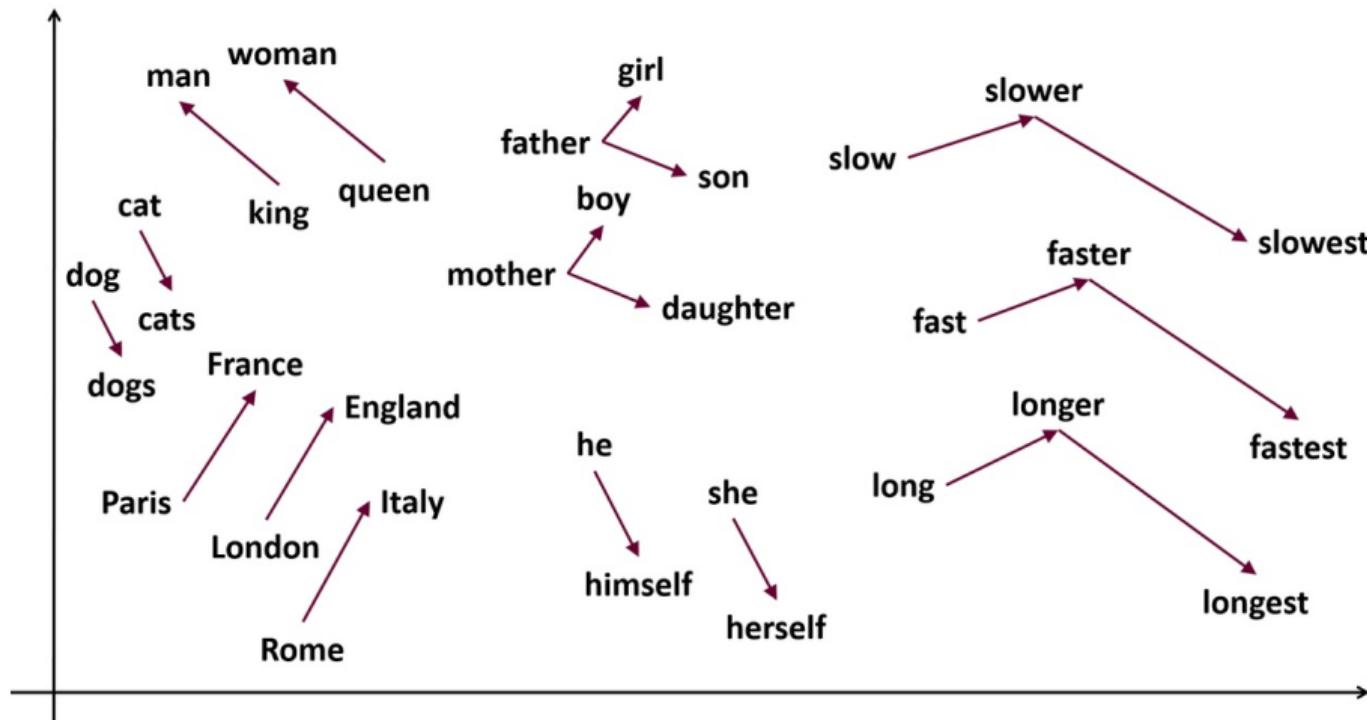
Word embeddings

- **Наша цель:** хотим научить компьютер понимать слова
- Идея! Давайте превратим наши слова в вектора размера d
- Какими интересными свойствами такие вектора могли бы обладать?

	Король	Тигр	Джедай
королевность	0.99	0.99	0.02	
мужественность	0.99	0.05	-0.5	
звериность	0.05	0.66	0.99	
вознеможность	0.7	0.93	-0.1	

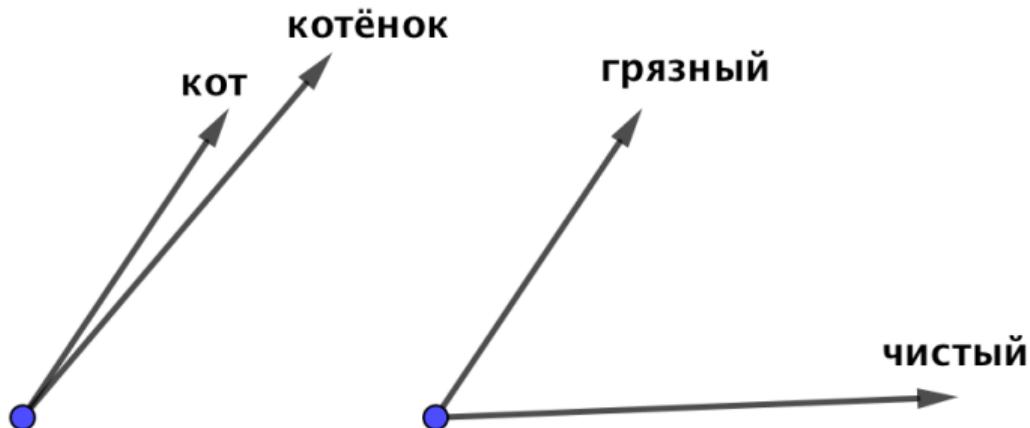
Свойство первое

- Модель улавливает семантические свойства слов



Свойство второе

- Модель понимает, где близкие по смыслу слова: кот, котёнок, кошка, тигр, лев, ...



Свойство третье

- Арифметика!



$$-\text{ } + \text{ } =$$
Two large, flat-colored silhouettes of human figures, one pink female and one blue male, standing side-by-side with their hands on their hips. They are positioned between the first image and the third image.



Томаш Миколов

- Звучит как магия, но так бывает
- В 2013 году такая модель предложена чешским аспирантом Томашем Миколовым
- После работал в Google, сейчас ушёл в Facebook



<https://arxiv.org/abs/1301.3781>

Основная идея

- Смысл слова сильно связан с соседними словами
- Если выкинуть слово, то оно должно хорошо восстанавливаться по представлениям соседних слов (контексту)
- Аналогично, по слову можно угадать его контекст
- Будем обучать представления слов, угадывая слово по контексту или контекст по слову

Постановка задачи

контекст
к слов до
к слов после

Вчера на обед Король Лев съел Пумбу

W1 W2

W0

W3 W4

- Слово **Лев** является контекстом слова **Король**
- Контекст — k слов до рассматриваемого и k после него
- Мы хотим предсказать контекст по центральному слову

Постановка задачи

контекст
к слов до
к слов после

Вчера на обед Король Лев съел Пумбу

W1 W2

W0

W3 W4

На каждое слово $w \in V$ (V — словарь всех слов) мы будем учить по два эмбеддинга:

- v_w , когда слово является центральным (w_0 на картинке)
- u_w , когда слово принадлежит контексту

Итого мы будем учить два набора векторов: $\theta = \left(\{u_w\}_{w \in V}, \{v_w\}_{w \in V} \right)$.

Постановка задачи

контекст
к слов до
к слов после

Вчера на обед Король Лев съел Пумбу

W1 W2 W0 W3 W4

Определим вероятность, что слово o (outside) окажется в контексте слова c (central):

$$\mathbb{P}_\theta(o \mid c) = \text{Softmax}_o \left(u_{w_1} \cdot v_c, \dots, u_{w_{|V|}} \cdot v_c \right) = \frac{\exp(u_o \cdot v_c)}{\sum_{w \in V} \exp(u_w \cdot v_c)}$$

В этой модели слова в контексте имеют категориальное распределение.

Метод максимального правдоподобия

Пусть текст состоит из T слов, $\{w_t\}_{t=1}^T$. Правдоподобие текста и его логарифм будут такими:

$$\mathcal{L}(\theta) = \prod_{t=1}^T \prod_{\substack{-k \leq j \leq k \\ j \neq 0}} \mathbb{P}_\theta(w_{t+j} | w_t) \rightarrow \max_\theta$$

$$\log \mathcal{L}(\theta) = \sum_{t=1}^T \sum_{\substack{-k \leq j \leq k \\ j \neq 0}} \log \mathbb{P}_\theta(w_{t+j} | w_t) \rightarrow \max_\theta$$

Мы хотим настроить вектора $\{u_w\}_{w \in V}, \{v_w\}_{w \in V}$ для слов так, чтобы $P_\theta(o | c)$ была высокой, если слово o встречается в контексте слова c .

Проблема

- В знаменателе Softmax есть суммирование по всем контекстуальным векторам $\{u_w\}_{w \in V} \dots$

Проблема

- В знаменателе Softmax есть суммирование по всем контекстуальным векторам $\{u_w\}_{w \in V} \dots$
- ...и по каждому из этих векторов градиент лосса ненулевой!
- Позже мы посмотрим на другие вероятностные модели текста, позволяющие избежать этой проблемы.

CBOW и skip-gram

- Пока мы рассматривали модель, где по центральному слову (c) мы предсказывали внешнее слово (o). Такая модель называется **skip-gram**:

$$\mathcal{L}(\theta) = \prod_{t=1}^T \prod_{\substack{-k \leq j \leq k \\ j \neq 0}} \text{Softmax}_{w_{t+j}}(u_{w_1} \cdot v_{w_t}, \dots, u_{w_{|V|}} \cdot v_{w_t})$$

- Также можно предсказывать центральное слово по внешним (точнее, по сумме их эмбеддингов). Этот вариант называется **continuous bag of words (CBOW)**:

$$\mathcal{L}(\theta) = \prod_{t=1}^T \text{Softmax}_{w_t}(u_o \cdot v_1, \dots, u_o \cdot v_{|V|}), \text{ где } u_o = \sum_{\substack{-k \leq j \leq k \\ j \neq 0}} u_{w_{t+j}}$$

- В CBOW те же проблемы со знаменателем Softmax, как и в skip-gram.

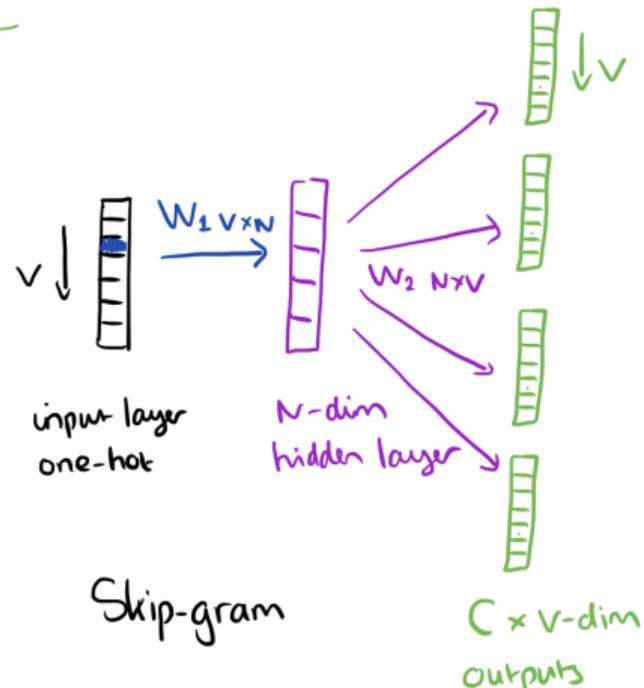
Обучение модели

Skip-gram

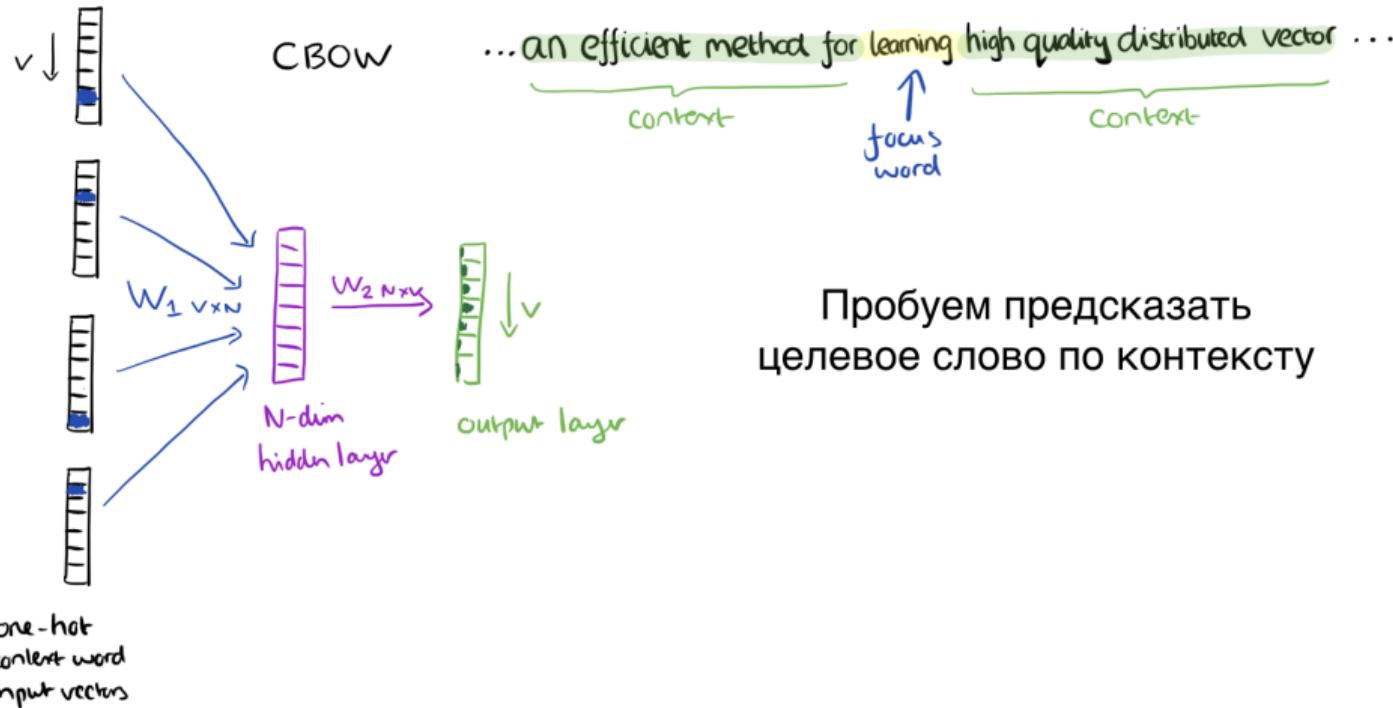
...an efficient method for learning high quality distributed vector ...



Пробуем предсказать
Контекст по целевому слову



CBOW



Обучение модели

- Трансформирует пространство текстов в d -мерное пространство векторов
- Для обучения требует много данных
- В матрице v будут записаны наши итоговые вектора для слов, в матрице u вектора для контекстов

Один шаг обучения в деталях

Ранее мы выводили такую формулу для максимизации правдоподобия:

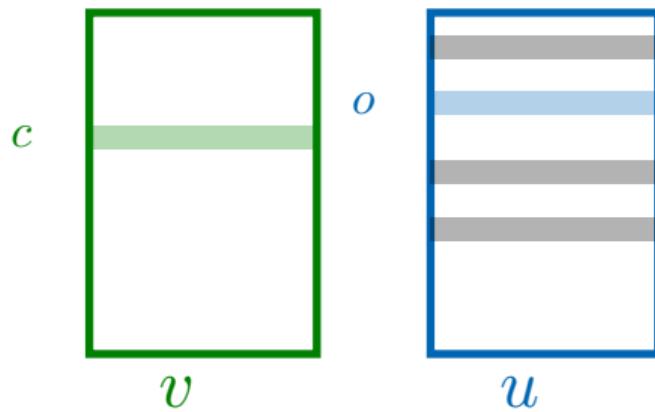
$$\sum_{t=1}^T \sum_{\substack{-k \leq j \leq k \\ j \neq 0}} \log \mathbb{P}_\theta(w_{t+j} | w_t) \rightarrow \max_\theta$$

Будем далее использовать сокращённые обозначения:

$$\sum_{c,o} \log \mathbb{P}_\theta(o | c) \rightarrow \max_\theta \quad - \sum_{c,o} \log \mathbb{P}_\theta(o | c) \rightarrow \min_\theta$$

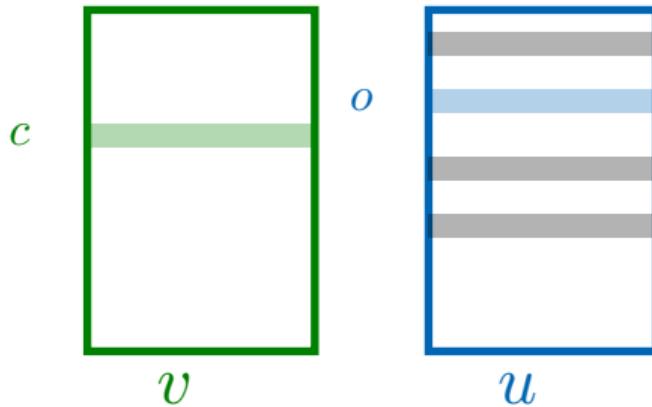
Один шаг обучения в деталях

$$J(\theta) = -\log \mathbb{P}_\theta(o | c)$$



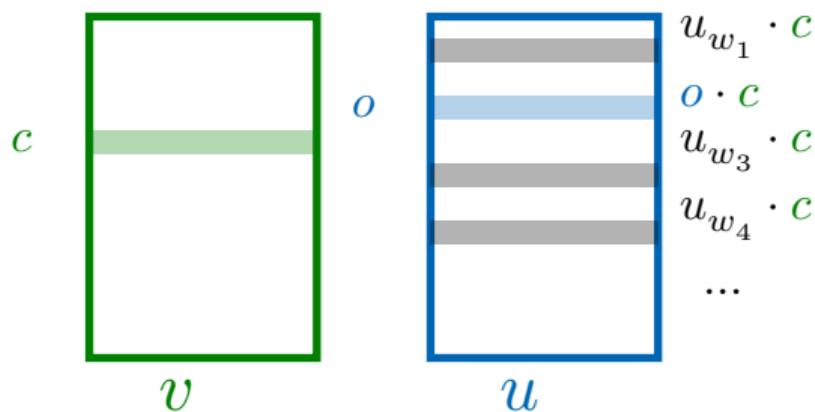
Один шаг обучения в деталях

$$J(\theta) = -\log \mathbb{P}_\theta(o \mid c) = -\log \frac{\exp(o \cdot c)}{\sum_{w \in V} \exp(u_w \cdot c)}$$



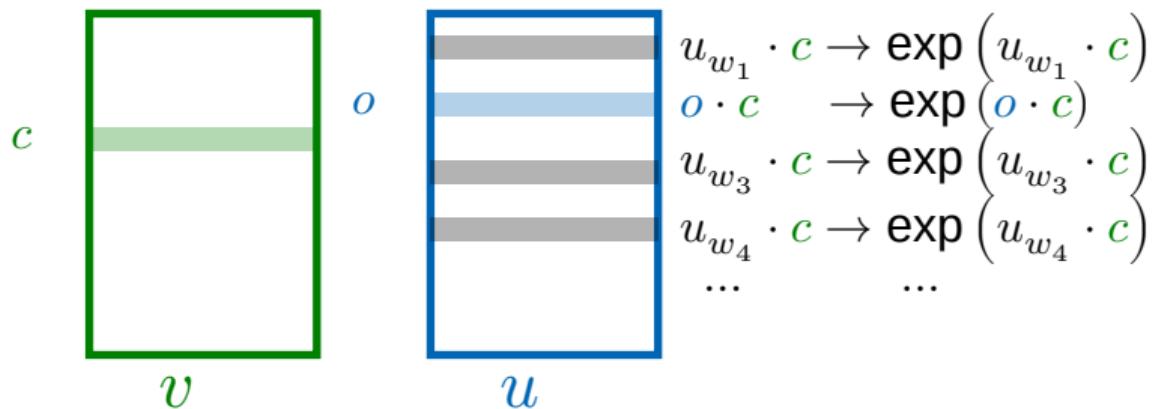
Один шаг обучения в деталях

$$J(\theta) = -\log \mathbb{P}_\theta(o \mid c) = -\log \frac{\exp(o \cdot c)}{\sum_{w \in V} \exp(u_w \cdot c)} = -o \cdot c + \log \sum_{w \in V} \exp(u_w \cdot c)$$



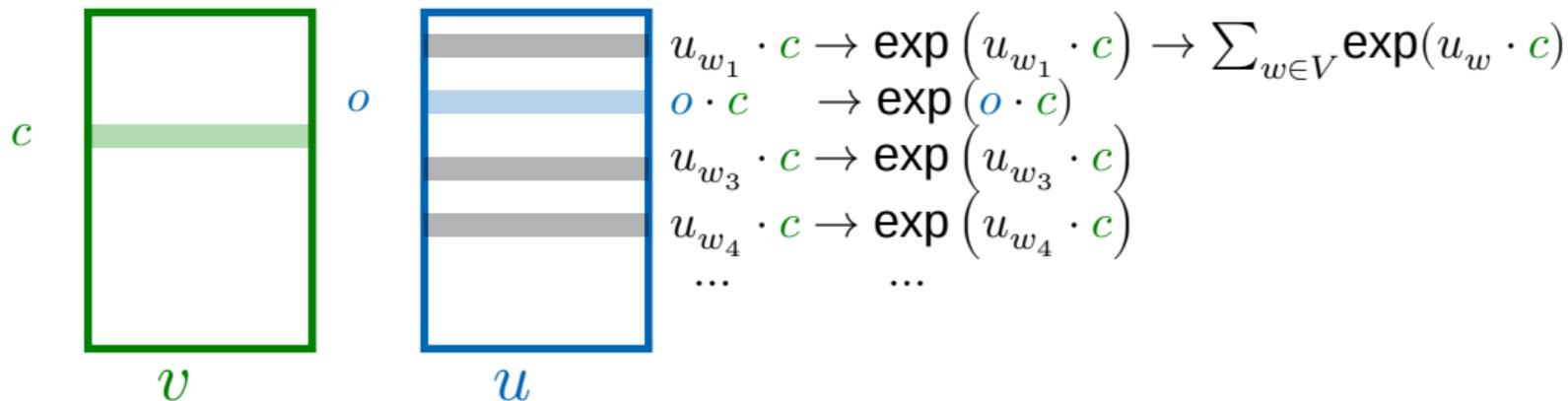
Один шаг обучения в деталях

$$J(\theta) = -\log \mathbb{P}_\theta(o \mid c) = -\log \frac{\exp(o \cdot c)}{\sum_{w \in V} \exp(u_w \cdot c)} = -o \cdot c + \log \sum_{w \in V} \exp(u_w \cdot c)$$



Один шаг обучения в деталях

$$J(\theta) = -\log \mathbb{P}_\theta(o \mid c) = -\log \frac{\exp(o \cdot c)}{\sum_{w \in V} \exp(u_w \cdot o)} = -o \cdot c + \log \sum_{w \in V} \exp(u_w \cdot c)$$



Один шаг обучения в деталях

Функция потерь:

$$J = -\log \mathbb{P}_\theta(o \mid c) = -\log \frac{\exp(o \cdot c)}{\sum_{w \in V} \exp(u_w \cdot c)} = -o \cdot c + \log \sum_{w \in V} \exp(u_w \cdot c)$$

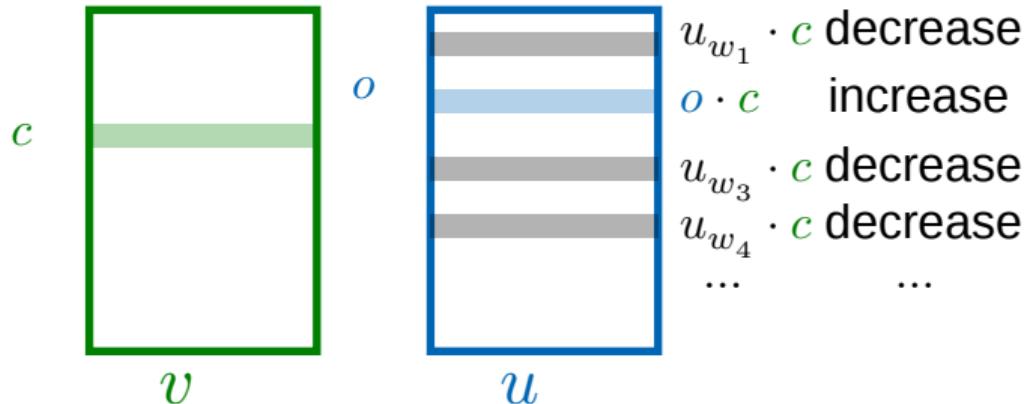
Шаг градиентного спуска:

$$\begin{aligned}c &= c - \gamma \cdot \frac{\partial J}{\partial c} \\u_w &= u_w - \gamma \cdot \frac{\partial J}{\partial u_w} \quad \forall w \in V\end{aligned}$$

https://lena-voita.github.io/nlp_course/word_embeddings.html

Один шаг обучения в деталях

$$J = -\log \mathbb{P}_\theta(o \mid c) = -\log \frac{\exp(o \cdot c)}{\sum_{w \in V} \exp(u_w \cdot c)} = -o \cdot c + \log \sum_{w \in V} \exp(u_w \cdot c)$$



Обновляем один c и каждый u_w , то есть всего $|V| + 1$ вектор
https://lena-voita.github.io/nlp_course/word_embeddings.html

Negative Sampling

- Нужно обновлять много параметров \Rightarrow медленное обучение
- Многие слова вместе не встречаются, поэтому большая часть вычислений избыточная
- Давайте максимизировать вероятность типичного контекста и минимизировать вероятность нетипичного контекста

Negative Sampling

- Правдоподобие для обычного skip-gram выглядело так:

$$\mathcal{L}(\theta) = \prod_{c,o} \frac{\exp(u_o \cdot v_c)}{\sum_{w \in V} \exp(u_w \cdot v_c)}$$

Ему соответствовала категориальная вероятностная модель на контекстах.

- Рассмотрим новую функцию правдоподобия:

$$\mathcal{L}(\theta) = \prod_{c,o} \left[\sigma(u_o \cdot v_c) \prod_{w \in \{w_{i_1}, \dots, w_{i_K}\}} (1 - \sigma(u_w \cdot v_c)) \right]$$

где $\{w_{i_1}, \dots, w_{i_K}\}$ — это случайные слова из словаря.

- Здесь вероятностная модель будет другой. **Какой?**

Negative Sampling

- Рассмотрим новую функцию правдоподобия:

$$\mathcal{L}(\theta) = \prod_{c,o} \left[\sigma(u_o \cdot v_c) \prod_{w \in \{w_{i_1}, \dots, w_{i_K}\}} (1 - \sigma(u_w \cdot v_c)) \right]$$

где $\{w_{i_1}, \dots, w_{i_K}\}$ — это случайные слова из словаря.

- Здесь вероятностная модель будет другой:
 - Слова, которые бывали контекстом o для слова c в настоящем тексте, попадают в этот контекст с вероятностью $\sigma(u_o \cdot v_c)$;
 - Случайные слова w попадают в этот контекст с вероятностью $1 - \sigma(u_w \cdot v_c)$.

Гиперпараметры

- Размер эмбеддинга (исторически 300, но варианты 100, 500 тоже возможно)
- Число наблюдений для негативного сэмплирования (для маленьких датасетов 15 – 20, для больших 2 – 5)
- Окно контекста (обычно 5 – 10)

https://lena-voita.github.io/nlp_course/word_embeddings.html

Полезные мысли про обучение

- В PyTorch довольно легко собрать свой собственный w2v и обучить его, но не стоит делать это. Ваша реализация не будет такой эффективной, как уже существующие специализированные реализации. За последние годы алгоритмы для обучения w2v претерпели существенную эволюцию.
- Реализация w2v из пакета gensim зачастую работает быстрее, чем модели, написанные в стандартных для нейросеток бэкэндах. Это происходит из-за многопоточности и разных умных оптимизаций тонких мест в обучении.

Полезные мысли про обучение

- При достаточно большом корпусе текстов можно не делать лемматизацию. Сетка сама поймёт по контексту, что слова близки и присвоит им похожие вектора.
- Если у вас специфическая задача, в которой встречается специфическая лексика, возьмите предобученную на большом корпусе сетку и дообучите её под свои нужды.
- w2v может выдавать эмбеддинги только для слов из заданного при обучении словаря. Этот минус можно попытаться побороть и получить другую модель, fasttext

Проблемы word2vec

- Не умеем работать с новыми словами, которых не было в нашем словаре при обучении
- Не закладываем никакой априорной информации о разных формах одного словаа
- Обучаемся на контекст слова, но всё ещё действуем в парадигме мешка слов, никак не учитываем порядок слов
- Не учитываем структуру слов, не умеем обрабатывать опечатки

Как это использовать и где
раздобыть?

Как это использовать

- Можно искать похожие слова
- Можно менять формы слов
- Можно искать определённые отношения
- Можно использовать как признаки для моделей
- Обучение w2v — аналог transfer learning, но обучение идёт на фиктивную задачу, разметка, сделанная вручную, не нужна

Something2vec

- Эмбеддинги (embeddings) — это сопоставление произвольной сущности (например, узла в графе или кусочка картинки) некоторому вектору.
- Любую последовательность можно представить в виде эмбеддинга
- Последовательность банковских транзакций
- Веб-сессии (последовательность перехода по сайтам)
- Графы взаимосвязей между пользователями
- Любая категориальная переменная: порядок, в котором турист посещал города; порядок, в котором юзер отранижировал сериалы и тп

something2vec



-



+



=



Где взять уже готовое (Google)

Google Code Archive

Search this site

Projects Search About

Project word2vec

Source

Issues Tool for computing continuous distributed representations of words.

Wikis

Downloads

Introduction

This tool provides an efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words. These representations can be subsequently used in many natural language processing applications and for further research.

Quick start

- Download the code: svn checkout <http://word2vec.googlecode.com/svn/trunk/>
- Run 'make' to compile word2vec tool
- Run the demo scripts: `./demo-word.sh` and `./demo-phrases.sh`
- For questions about the toolkit, see <http://groups.google.com/group/word2vec-toolkit>

Project Information

The project was created on Jul 30, 2013.

- License: Apache License 2.0
- 945 stars
- svn-based source control

Labels:

NeuralNetwork MachineLearning
NaturalLanguageProcessing WordVectors
Google

Гуглowsкая модель для английского языка:
<https://code.google.com/archive/p/word2vec/>

Где взять уже готовое (проект RusVectōrēs)

Модели

В настоящий момент вы можете скачать следующие модели (жирным выделены модели, доступные для использования в веб-интерфейсе):

Таблицу можно (и нужно) пролистывать по горизонтали!

Постоянный идентификатор	Скачать	Корпус	Размер корпуса	Объём словаря	Частотный порог	Тэгсет	Алгоритм	Размерность вектора	Размер окна
a_upos_skipgram_300_2_2018	331 Мбайт	Тайга	почти 5 миллиардов слов	237 255	200	Universal Tags	Continuous Skipgram	300	2
corpora_upos_skipgram_300_5_2018	191 Мбайт	НКРЯ	250 миллионов слов	195 071	20	Universal Tags	Continuous Skipgram	300	5
ikiuruscorpora_upos_skipgram_300_2_2018	376 Мбайт	НКРЯ и Википедия за декабрь 2017	600 миллионов слов	384 764	40	Universal Tags	Continuous Skipgram	300	2
rs_upos_cbow_600_2_2018	547 Мбайт	Русскозычные новости, с сентября 2013 до ноября 2016	почти 5 миллиардов слов	289 191	200	Universal Tags	Continuous Bag-of-Words	600	2
araneum_upos_skipgram_300_2_2018	192 Мбайта	Araneum	около 10 миллиардов слов	196 620	400	Universal Tags	Continuous Skipgram	300	2
araneum_none_fasttextcbow_300_5_2018	1 Гбайт	Araneum	около 10 миллиардов слов	195 782	400	Нет	fastText CBOW (3.-5-граммы)	300	5
araneum_none_fasttextskipgram_300_5_2018	675 Мбайт	Araneum	около 10 миллиардов слов	195 782	400	Нет	fastText Skipgram	300	5

Куча разных моделей для русского языка: <https://rusvectores.org/ru/models/>

Свойства word2vec



Частотность слова

- Высокая Средняя Низкая

НКРЯ и Wikipedia

- чай noun 0.56
- пиво noun 0.56
- самогон noun 0.56
- лимонад noun 0.53
- напиток noun 0.53



Частотность слова

- Высокая Средняя Низкая

НКРЯ и Wikipedia

- преданность noun 0.38
- доброта noun 0.37
- нежность noun 0.37
- упование noun 0.35
- умиление noun 0.35

<https://rusvectores.org/ru/calculator>

Свойства word2vec

Результат обучения векторных представлений сильно зависит от коллекции документов. Могут возникать неожиданные артефакты.

Википедия

most_similar(россия)
российский 0.5653642416
рф 0.523694574833
украина 0.492026507854
ссср 0.473026573658
финляндия 0.464367419481
most_similar(тролль)
муметь 0.717674195766
гоблин 0.559770524502
великан 0.557757973671
злобный 0.55741250515
гном 0.554968833923

Луркоморье

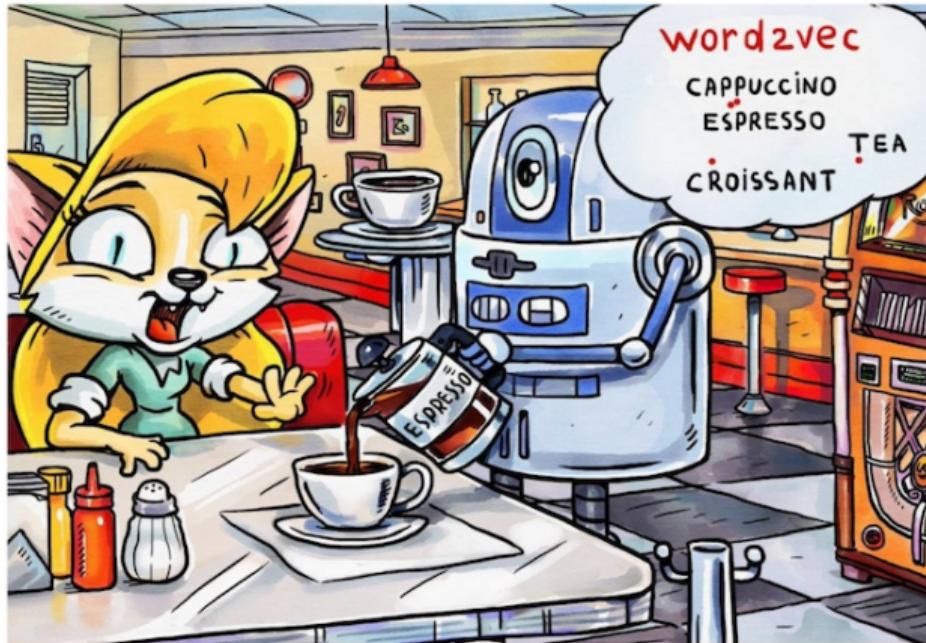
most_similar(россия)
беларусь 0.645048737526
европа 0.622894406319
украина 0.622316598892
рашка 0.619276404381
германия 0.609378278255
most_similar(тролль)
троллинг 0.725703835487
троль 0.660580933094
лжец 0.582996308804
проводокатор 0.57004237175
толстый 0.568691492081

Модель-сексист

```
model.most_similar(u'интеллектуал')
[('моралист', 0.7139864563941956),
('теоретик', 0.6941959857940674),
('литератор', 0.6819325089454651)]  
  
model.most_similar(u'интеллектуалка')
[('бездельница', 0.6617184281349182),
('бунтарка', 0.6578608751296997),
('дилетантка', 0.6419748663902283)]  
  
  
model.most_similar(positive=[u'интеллектуал', u'женщина'], negative=[u'мужчина'])
[('знаменитость', 0.49774518609046936),
('поп-звезда', 0.4860984981060028),
('элита', 0.48200151324272156)]  
  
model.most_similar(positive=[u'ум', u'женщина'], negative=[u'мужчина'])
[('любовь', 0.43659064173698425),
('душа', 0.4330841302871704),
('внешность', 0.4280041456222534)]  
  
model.most_similar(positive=[u'гений', u'женщина'], negative=[u'мужчина'])
[('букашка', 0.4793989062309265),
('химера', 0.4589369595050812),
('душонка', 0.4547439217567444)]
```

<https://nikolenko.livejournal.com/267442.html>

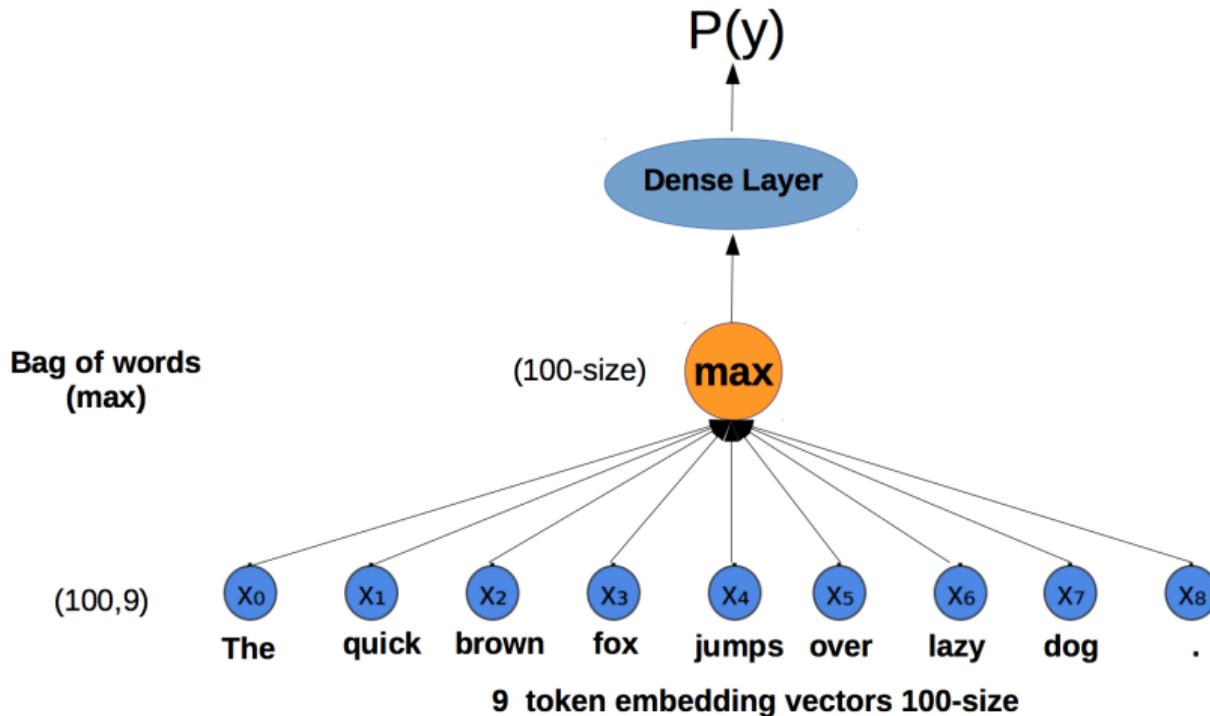
Word2vec — всего лишь модель



- Espresso? But I ordered a cappuccino!
- Don't worry, the cosine distance between them is so small that they are almost the same thing.

Свёрточные нейросети для текстов

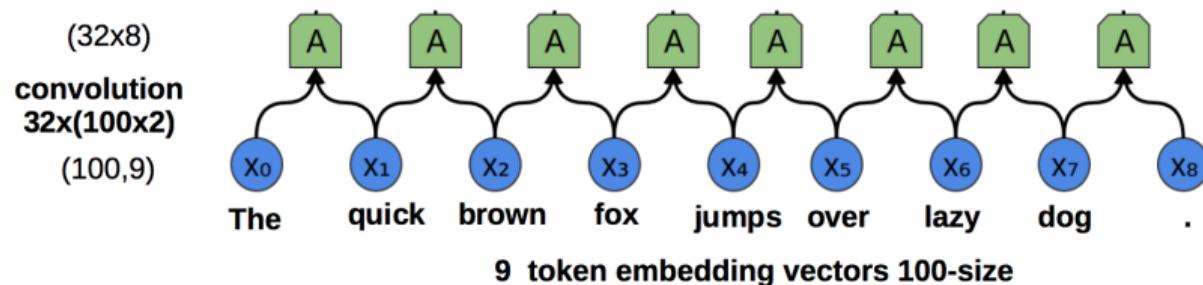
Нейросеть для текстов



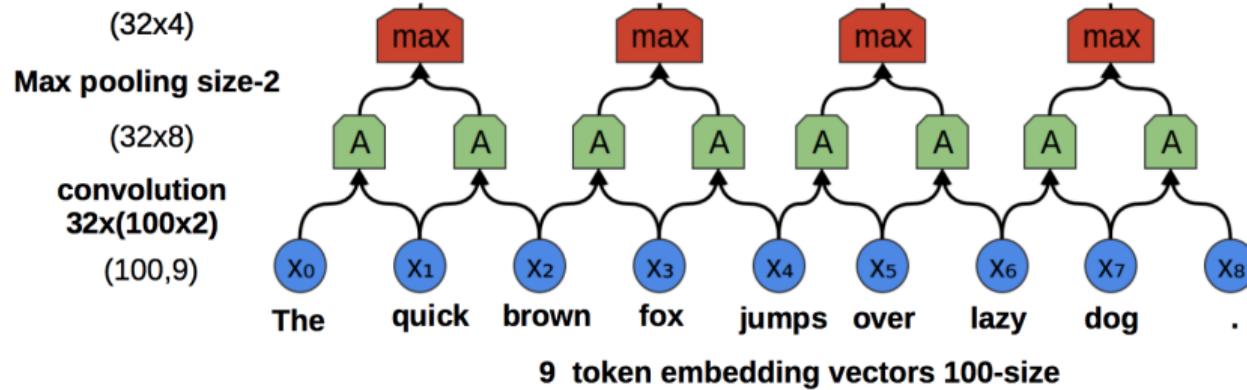
Проблемы

- Теряем информацию о порядке слов
- Решить эту проблему можно, обучив эмбеддинги для биграмм, триграмм и т.д., но это расширит пространство признаков
- Можно решить эту проблему с помощью рекуррентных нейросеток, о них мы будем говорить в следующий раз
- Можно решить эту проблему с помощью свёрточного слоя

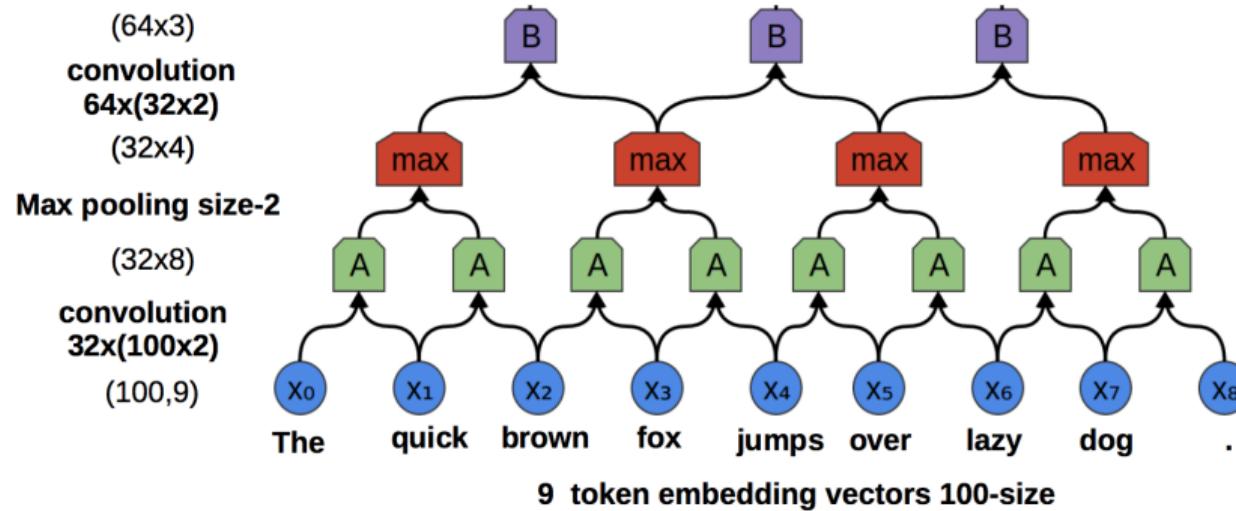
Свёрточная сетка



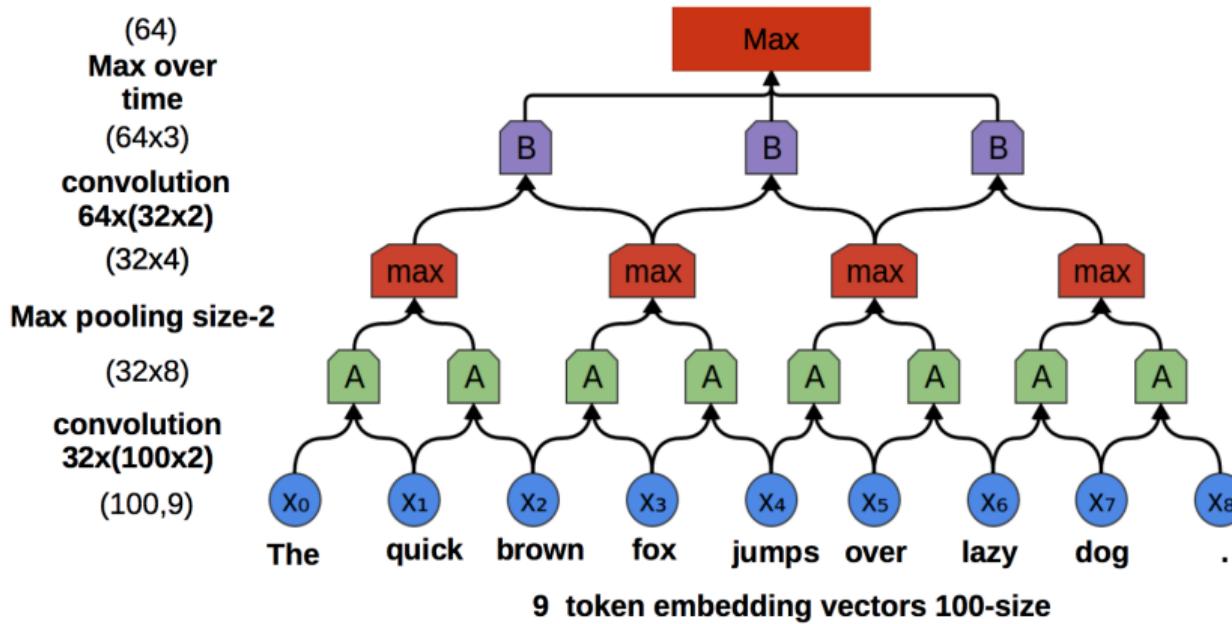
Свёрточная сеть



Свёрточная сетка



Свёрточная сетка



Свёрточная сетка

