

Автоматическая обработка текстов Word Embeddings

Лекция 2

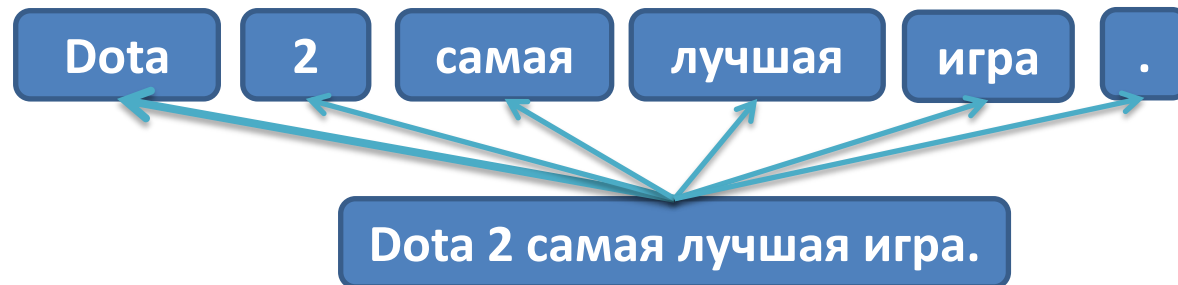
Емельянов А. А.
login-const@mail.ru

Зачем нужны представления слов?

Dota 2 самая лучшая игра.

Текст

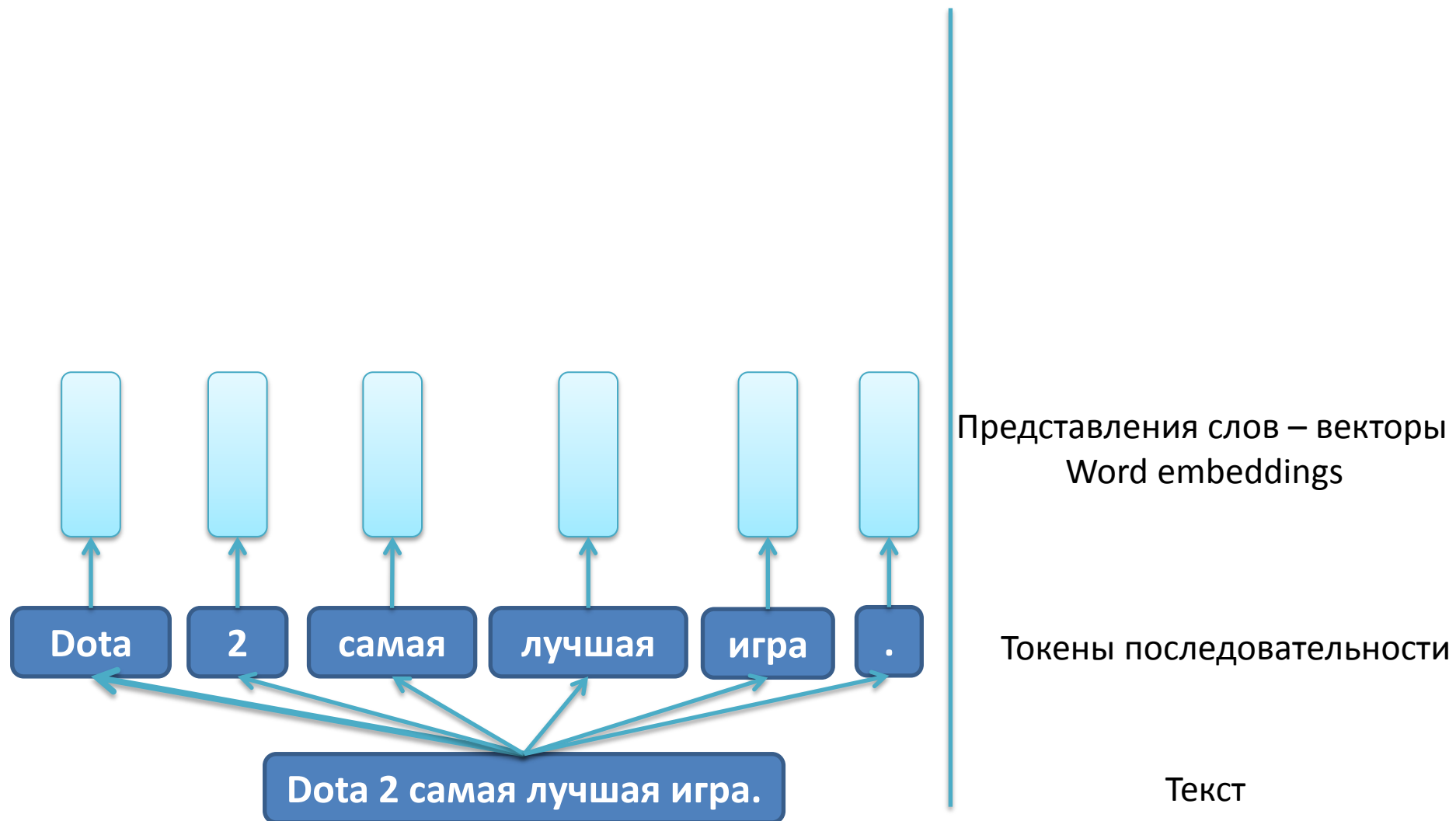
Зачем нужны представления слов?



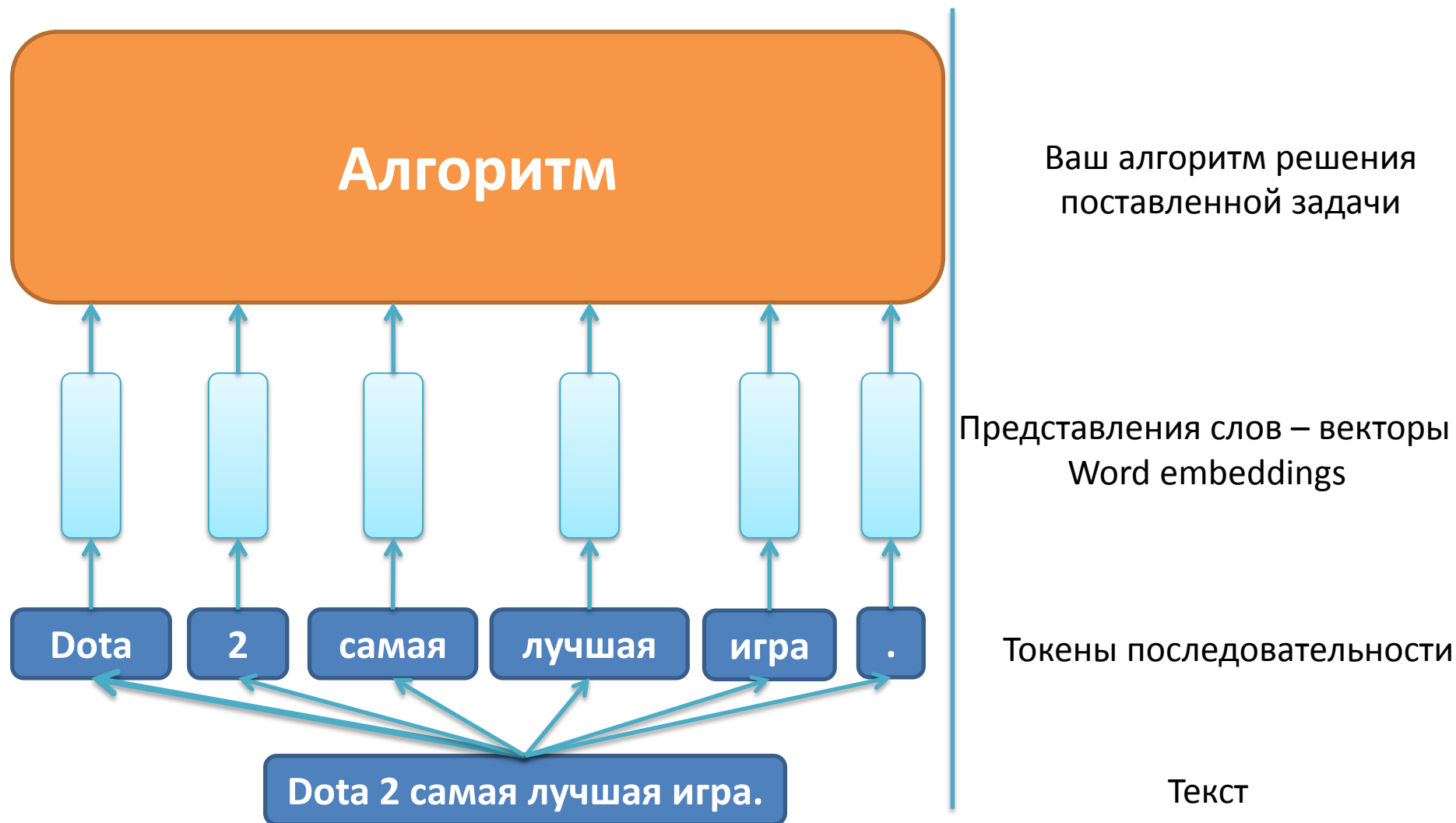
Токены последовательности

Текст

Зачем нужны представления слов?



Зачем нужны представления слов?



- **Коллокацией** называется словосочетание, имеющее признаки синтаксически и семантически целостной единицы, в котором выбор одного из компонентов осуществляется по смыслу, а выбор второго зависит от выбора первого (например, «**ставить условия**» — выбор глагола ставить определяется традицией и зависит от существительного условия, при слове «**предложение**» будет другой глагол — «**вносить**»).

Статистический подход: биграммы

- Топ биграмм обычно не то, что надо ☹

№	Словосочетание	Документы	Частота
1	<u>и не</u>	22732	201352
2	<u>и в</u>	27048	193983
3	<u>потому что</u>	14926	117401
4	<u>я не</u>	10675	113767
5	<u>у меня</u>	9734	97102
6	<u>может быть</u>	16086	96065
7	<u>то что</u>	17195	95251
8	<u>что он</u>	11786	92743
9	<u>не было</u>	13196	92729

Биграммы с учетом частей речи

A
на расстоянии 1 от S

№	Вхождения	Документы	Фрагмент
1	19402	6104	крайней мере
2	18152	2791	российской федерации
3	12164	6528	настоящее время
4	11348	4160	должны были
5	11045	6067	последнее время
6	9720	2893	молодой человек

S
на расстоянии 1 от S

№	Вхождения	Документы	Фрагмент
1	45631	9556	а потом
2	21563	10492	том числе
3	17401	5932	друг друга
4	15362	6214	точки зрения
5	14925	5242	конце концов
6	12616	4597	т п

V
на расстоянии 1 от S

№	Вхождения	Документы	Фрагмент
1	8583	4415	три года
2	7404	2919	следующий день
3	7122	3044	три дня
4	6615	2851	было уже
5	5716	3262	данном случае
6	5345	2415	был уже

NUM
на расстоянии 1 от S

№	Вхождения	Документы	Фрагмент
1	14054	5071	несколько раз
2	12787	4376	несколько дней
3	10561	5359	два года
4	9248	5078	несколько лет
5	8583	4415	три года
6	8123	2852	несколько минут

Биграммы со словом большой

большой

№	Вхождения	Документы	Фрагмент
1	14299	4839	больше не
2	12725	6087	больше чем
3	10226	4365	и больше
4	9912	4733	с большим
5	8307	3992	больше всего
6	7978	3701	еще больше
7	7434	3772	все больше
8	5595	3039	в большом
9	5465	2852	больше и
10	5319	2862	не больше

большой на расстоянии 1 от S

№	Вхождения	Документы	Фрагмент
1	4372	2678	большая часть
2	2872	1952	большую часть
3	1933	1417	большое количество
4	1692	1084	большое значение
5	1650	1058	больше того
6	1518	1066	больше нет
7	1190	763	большую роль
8	1164	863	большим трудом
9	1130	866	большие деньги
10	905	534	большого театра

Биграммы со словом огромный

огромный

№	Вхождения	Документы	Фрагмент
1	1809	1266	с огромным
2	1437	1105	огромное количество
3	1254	933	в огромном
4	924	722	с огромными
5	889	711	с огромной
6	723	579	огромное значение
7	717	585	в огромной
8	579	508	в огромных
9	436	382	на огромном
10	417	326	огромную роль

огромный

на расстоянии 1 от S

№	Вхождения	Документы	Фрагмент
1	1437	1105	огромное количество
2	723	579	огромное значение
3	417	326	огромную роль
4	321	221	огромное большинство
5	311	263	огромные деньги
6	266	237	огромном количестве
7	262	219	огромное влияние
8	245	227	огромного количества
9	222	199	огромная толпа
10	206	172	огромное число

Плюсы и минусы биграммного подхода

Плюсы и минусы биграммного подхода

- **Плюсы:**
 - + простота

Плюсы и минусы биграммного подхода

- **Плюсы:**
 - + простота
 - + хорошо работает для фиксированных фраз

Плюсы и минусы биграммного подхода

- **Плюсы:**
 - + простота
 - + хорошо работает для фиксированных фраз
- **Минусы:**

Плюсы и минусы биграммного подхода

- **Плюсы:**

- + простота
- + хорошо работает для фиксированных фраз

- **Минусы:**

- плохо работает для слов, не обязательно стоящих рядом:

стучать во все **двери**

стучать во все возможные **двери**

в **дверь** постучали

в **дверь** купе постучали

постучал в новую **дверь**

не ошибся **дверью** и **постучал**

Распределение расстояний между словами

- Посчитаем по выборке среднее расстояние (со знаком) между словами и его дисперсию:

$$\mu = \frac{1}{6} (3 + 4 - 1 - 2 + 3 + 2) = 1.5$$

$$\sigma^2 = \frac{\sum_{i=1}^n (d_i - \mu)^2}{n - 1} \approx 2.42$$

- Чем меньше σ , тем больше слова похожи на коллокацию.

Распределение расстояний между словами

σ	μ	частота	w_1	w_2
0,43	0,97	11657	New	York
0,48	1,83	24	previous	games
0,15	2,98	46	minus	points
0,49	3,87	131	hundreds	dollars
4,03	0,44	36	editorial	Atlanta
4,03	0,00	78	ring	New
3,96	0,19	119	point	hundredth
3,96	0,29	106	subscribers	by
1,07	1,45	80	strong	support
1,13	2,57	7	powerful	organizations
1,01	2,00	112	Richard	Nixon
1,05	0,00	10	Garrison	said

$$PMI(w_1, w_2) = \log_2 \frac{P(w_1 w_2)}{P(w_1)P(w_2)}$$

- P – частота слова или биграммы.
- Оценивает независимость совместного появления пары слов.
- Значения величины зависят от размеров корпуса.
- Завышает значимость редких словосочетаний.
- Решение: порог по частоте.
- Выделяет терминологические словосочетания.

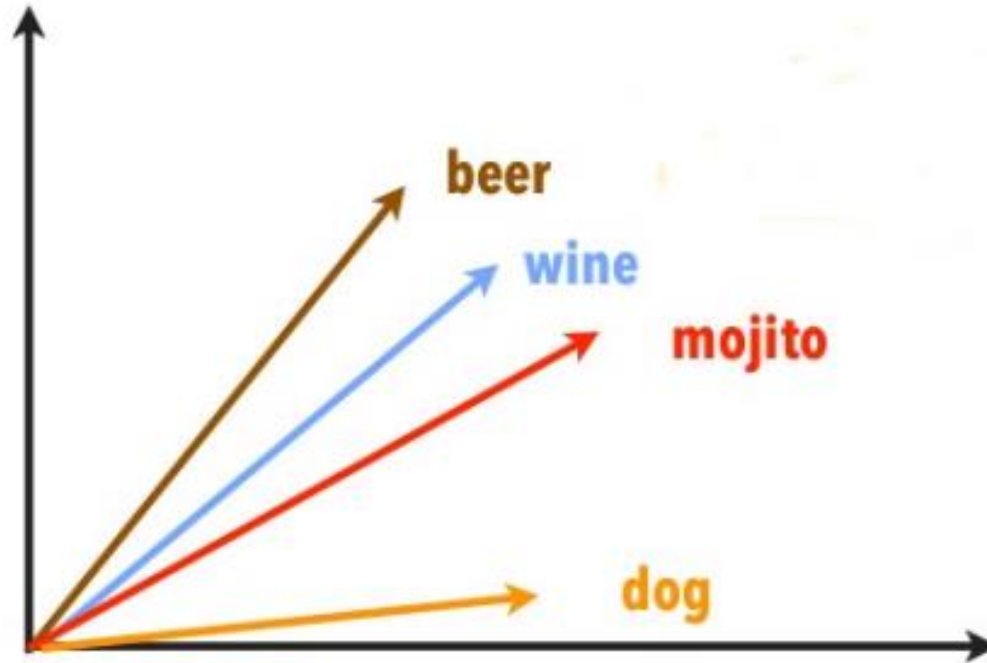
$$PPMI(w_1, w_2) = \max(PMI(w_1, w_2), 0)$$

— положительная поточечная взаимная информация.

- **Задача:** найти слова, синтаксически и/или семантически «ближайшие» к данному слову.

Векторное представление слов

- **Идея:** каждое слово представлять вектором в некотором пространстве R^n .



One-Hot Encoding

- Предположим, что слова это дискретные «символы»: **motel**, **hotel**.
- Нумеруем все слова в словаре.
- Получаем для каждого слова вектор с единицей в позиции с номером слова и нули в остальных местах.

motel = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

hotel = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

One-Hot Encoding - проблемы

- Пример: при веб запросе «**Moscow motel**» мы хотим также получить в качестве результата «**Moscow hotel**».

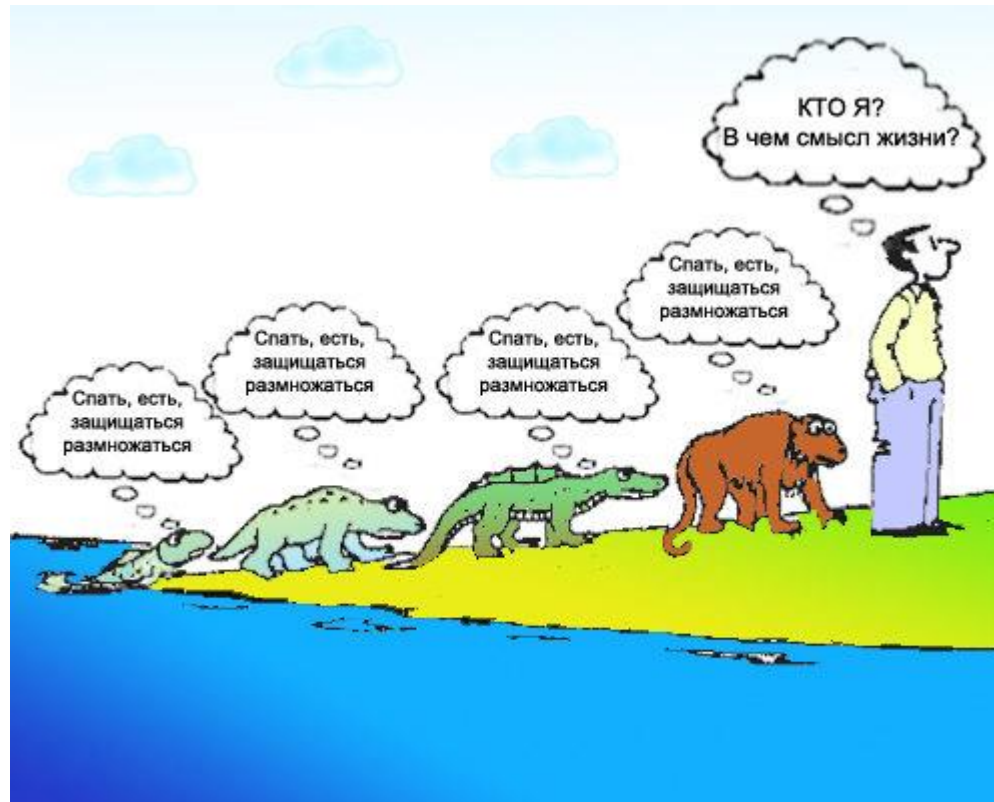
- Однако

motel = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

hotel = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

- Любые два таких вектора ортогональны.
- В таком представлении нет *естественной* похожести слов.
- Данные вектора не содержат информации о смыслах слов.

Что такое смысл?



Что такое смысл?

- Что такое бардюк?

Что такое смысл?

- Что такое бардюк?
- Он подал ей бокал бардюка.
- Бардюк подают к блюдам из говядины.
- Ноги у него заплетались, а лицо горело от выпитого бардюка.
- Виноград сорта бардюк выращивают в Австралии.
- К простому ужину из хлеба и сыра я взял бутылку отличного местного бардюка.
- Напитки были прекрасны: кроваво-красный бардюк и легкое белое рейнское.

Дистрибутивная семантика

- Что еще может стоять в данном месте?
- На столе стояла бутылка _____. (1)
- Все любят _____. (2)
- Не употребляйте _____ перед вождением. (3)
- Мы делаем _____ из кукурузы. (4)

Дистрибутивная семантика

- Что еще может стоять в данном месте?
- На столе стояла бутылка _____. (1)
- Все любят _____. (2)
- Не употребляйте _____ перед вождением. (3)
- Мы делаем _____ из кукурузы. (4)

	(1)	(2)	(3)	(4)	...
бардюк	1	1	1	1	
моторное масло	1	0	0	0	
вино	1	1	1	0	
кричать	0	0	0	0	
выборы	0	1	0	0	

Дистрибутивная семантика

- Что еще может стоять в данном месте?
- На столе стояла бутылка _____. (1)
- Все любят _____. (2)
- Не употребляйте _____ перед вождением. (3)
- Мы делаем _____ из кукурузы. (4)

	(1)	(2)	(3)	(4)	...
бардюк	1	1	1	1	
моторное масло	1	0	0	0	
вино	1	1	1	0	
кричать	0	0	0	0	
выборы	0	1	0	0	

- Верно ли, что слова, встречающиеся в одинаковых контекстах, имеют похожие смыслы?

Дистрибутивная семантика

- Верно ли, что слова, встречающиеся в одинаковых контекстах, имеют похожие смыслы?



- Дистрибутивная гипотеза (J.R.Firth, 1957).

You shall know a word by the company it keeps!

Дистрибутивная семантика

doc#50340	так Остро ощущает, когда он влюблен. Эта	компульсия	может существовать и при отсутствии влюбленности
doc#336729	психологии оракул гороскоп такое понятие —	компульсия	— гороскоп притяжение к оракулу, реализующееся
doc#878536	(обсессии); </p><p> навязчивое поведение (компульсия); </p><p> оппозиционное поведение; </p><p>
doc#1000748	характерна другая сила тяги, так называемая «	компульсия	». </p><p> - И что означает это слово в применении
doc#1000748	учителей очень образно описывал, что такое «	компульсия	». Это влечение, сравнимое с жизненно важными
doc#1369221	обеспечиваются равновесие, гомеостазис. </p><p> Иногда	компульсия	лучше устраняется посредством ее «взрыва
doc#2553333	борьбы с ними. </p><p> Навязчивое влечение (компульсия) — стремление, вопреки разуму, воле и чувствам
doc#3060833	есть субъективный компонент — влечение, или	компульсия	, и объективный — ритуал (вызванные влечением
doc#3575480	рука поднимается вверх и запускается ваша	компульсия	, само ощущение будет буквально утягивать
doc#3575480	направлении. Не то, чтобы у вас исчезла	компульсия	, просто у вас появляется компульсия быть
doc#3575480	исчезла компульсия, просто у вас появляется	компульсия	быть более таким, каким вы хотите быть.
doc#4796843	вещи, которые усиливает эту компульсию, и	компульсия	потеряет всю свою силу. Сама компульсия
doc#4796843	компульсия потеряет всю свою силу. Сама	компульсия	это только то, что лежит на поверхности
doc#4796843	их основе тревога или нечто подавленное.	Компульсия	является защитным механизмом против чувства
doc#4796843	жизни тревогу или депрессию (Чаще всего	компульсия	приводится в действие тревогой-беспокойством
doc#4796843	начните прорабатывать эти чувства. Скоро ваша	компульсия	, независимо от того что вы делаете, уйдет
doc#4796843	непомерную, но полезную службу, которую	компульсия	выполняет для вас. Поблагодарите ее вовлекая
doc#4796843	После того как я обработал эту тревогу,	компульсия	к курению никогда больше не возвращалась
doc#4900615	максимально успешного лечения заболеваний] </p><p>	КОМПУЛЬСИЯ	compulsion [непреодолимое побуждение совершать
doc#5703937	родственными». Эти близнецы — навязчивость (компульсия) и торможение — знакомы каждому, кто испытывал

Явное представление слов контекстами

- Для словаря V_w и множества контекстов V_c построим разреженную матрицу $M_{[i,j]} = f(w_i, c_j)$ размера $|V_c| \times |V_w|$.
- Элемент $f(w_i, c_j)$ будет описывать связь слова w_i с контекстом c_j .

	c_1	c_2	...	$c_{ V_c }$
w_1	f_{11}	f_{12}		$f_{1 V_c }$
w_2	f_{21}	f_{22}		$f_{2 V_c }$
...				
$w_{ V_w }$	$f_{ V_w 1}$	$f_{ V_w 2}$		$f_{ V_w V_c }$

Явное представление слов контекстами

- Для словаря V_w и множества контекстов V_c построим разреженную матрицу $M_{[i,j]} = f(w_i, c_j)$ размера $|V_c| \times |V_w|$.
- Элемент $f(w_i, c_j)$ будет описывать связь слова w_i с контекстом c_j .
- Как определить $f(w_i, c_j)$?

Явное представление слов контекстами

- Для словаря V_w и множества контекстов V_c построим разреженную матрицу $M_{[i,j]} = f(w_i, c_j)$ размера $|V_c| \times |V_w|$.
- Элемент $f(w_i, c_j)$ будет описывать связь слова w_i с контекстом c_j .
- Как определить $f(w_i, c_j)$?
 - $\#(w, c)$

Явное представление слов контекстами

- Для словаря V_w и множества контекстов V_c построим разреженную матрицу $M_{[i,j]} = f(w_i, c_j)$ размера $|V_c| \times |V_w|$.
- Элемент $f(w_i, c_j)$ будет описывать связь слова w_i с контекстом c_j .
- Как определить $f(w_i, c_j)$?
 - $\#(w, c)$
 - $P(w, c) = \#(w, c), (w, c) \in D$ – наблюдаемые пары (слово, контекст), всего пар $|D|$.

Явное представление слов контекстами

- Для словаря V_w и множества контекстов V_c построим разреженную матрицу $M_{[i,j]} = f(w_i, c_j)$ размера $|V_c| \times |V_w|$.
- Элемент $f(w_i, c_j)$ будет описывать связь слова w_i с контекстом c_j .
- Как определить $f(w_i, c_j)$?
 - $\#(w, c)$
 - $P(w, c) = \#(w, c), (w, c) \in D$ – наблюдаемые пары (слово, контекст), всего пар $|D|$.
 - $PMI(w, c)$

Явное представление слов контекстами

- Для словаря V_w и множества контекстов V_c построим разреженную матрицу $M_{[i,j]} = f(w_i, c_j)$ размера $|V_c| \times |V_w|$.
- Элемент $f(w_i, c_j)$ будет описывать связь слова w_i с контекстом c_j .
- Как определить $f(w_i, c_j)$?
 - $\#(w, c)$
 - $P(w, c) = \#(w, c), (w, c) \in D$ – наблюдаемые пары (слово, контекст), всего пар $|D|$.
 - $PMI(w, c)$
 - $PPMI(w, c)$

Оценка близости между векторами

- Косинусная мера близости:

$$\cos(u,v) = \frac{uv}{\|u\|_2 \|v\|_2} = \frac{\sum_i u_i v_i}{\sqrt{\sum_i u_i^2} \sqrt{\sum_i v_i^2}}$$

- Мера Жаккара:

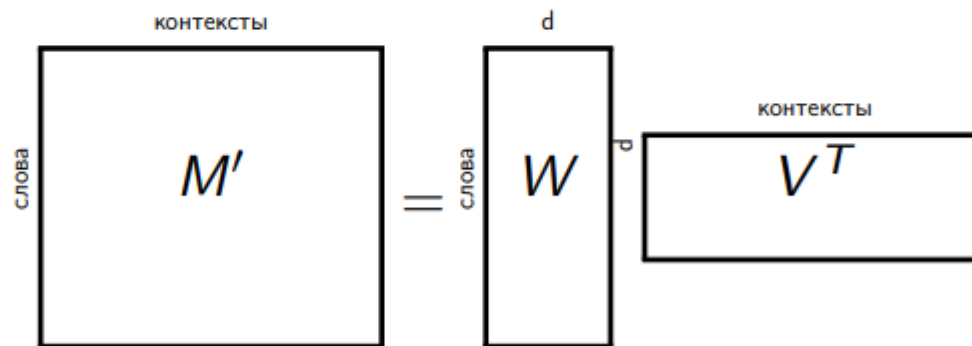
$$jc(u,i) = \frac{\sum_i \min(u_i, v_i)}{\sum_i \max(u_i, v_i)}$$

Уменьшение размерности

- С векторами такого размера работать неудобно.
- Будем строить векторы размерности $N \ll |V_c|$.
- **Факторизация** матрицы терм-контекст:

$$M' = W \times V^T, W \in \mathbb{R}^{V_w \times d}, V \in \mathbb{R}^{V_c \times V_d}$$

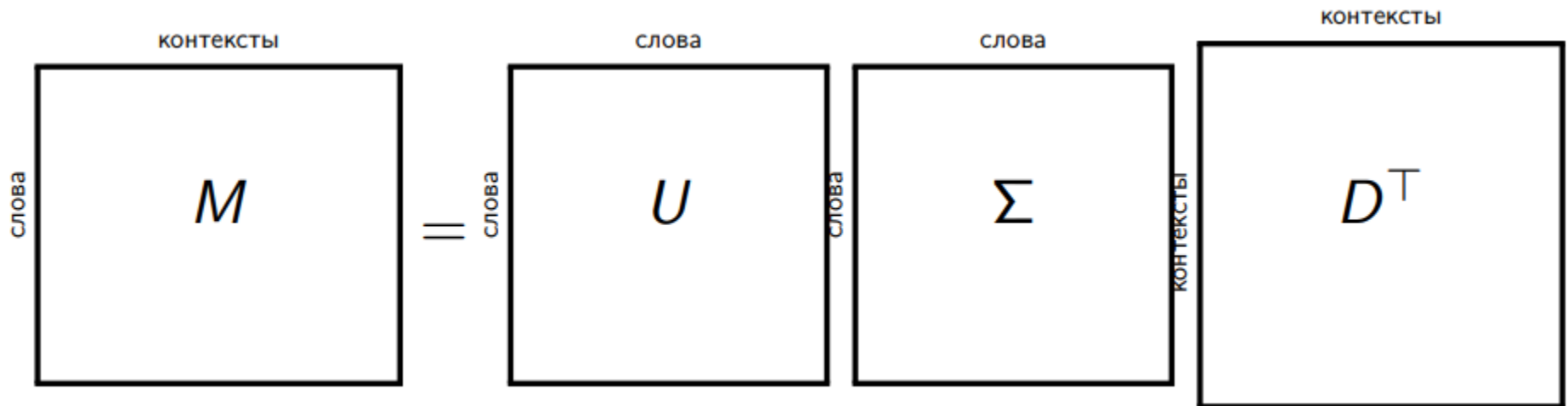
M' – лучшее приближение ранга d к M по L_2 .



Уменьшение размерности

- Сингулярное разложение матрицы слово-контекст $M \in R^{V_w \times V_c}$:

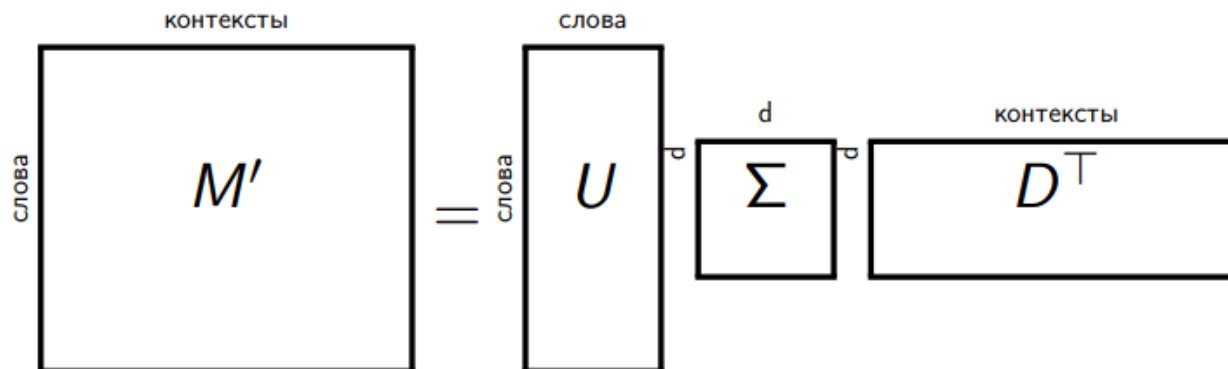
$$M = U \Sigma D^T$$



Уменьшение размерности

- Аппроксимация ранга d матрицы слово-контекст $M \in R^{V_w \times V_c}$:

$$M'_d = U_d \Sigma_d D_d^\top$$



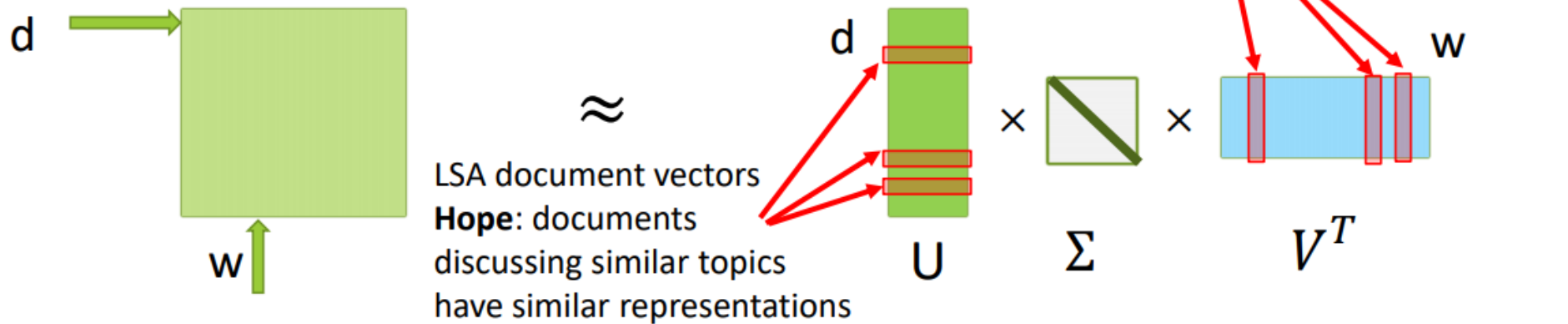
- Искомое разложение:

$$W = U_d \sqrt{\Sigma_d}, V^\top = \sqrt{\Sigma_d} D_d^\top$$

Латентно семантический анализ (1988)

X - document-term co-occurrence matrix

$$X \approx \hat{X} = U \Sigma V^T$$

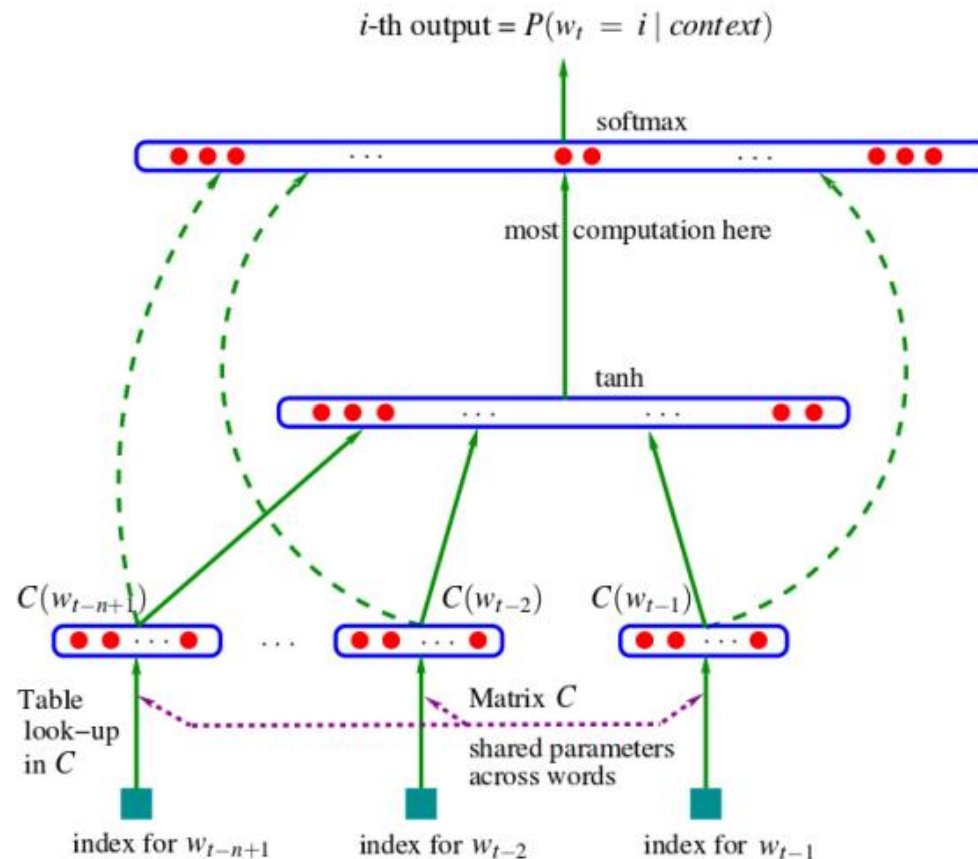


Латентно семантический анализ (1988)

- **Фактически** — применение SVD к матрице «документ – терм»
- **Возможности метода:**
 - оценка близости документов
 - оценка близости термов
 - кластеризация документов
 - оценка близости запроса и документа
- **Недостатки:**
 - низкая скорость
 - нет вероятностных предположений о распределении.

Неявное представление слов контекстами

- Другой способ получения векторного представления — нейронная сеть. Архитектуры нейронных сетей могут быть как последовательными, так и рекуррентными.

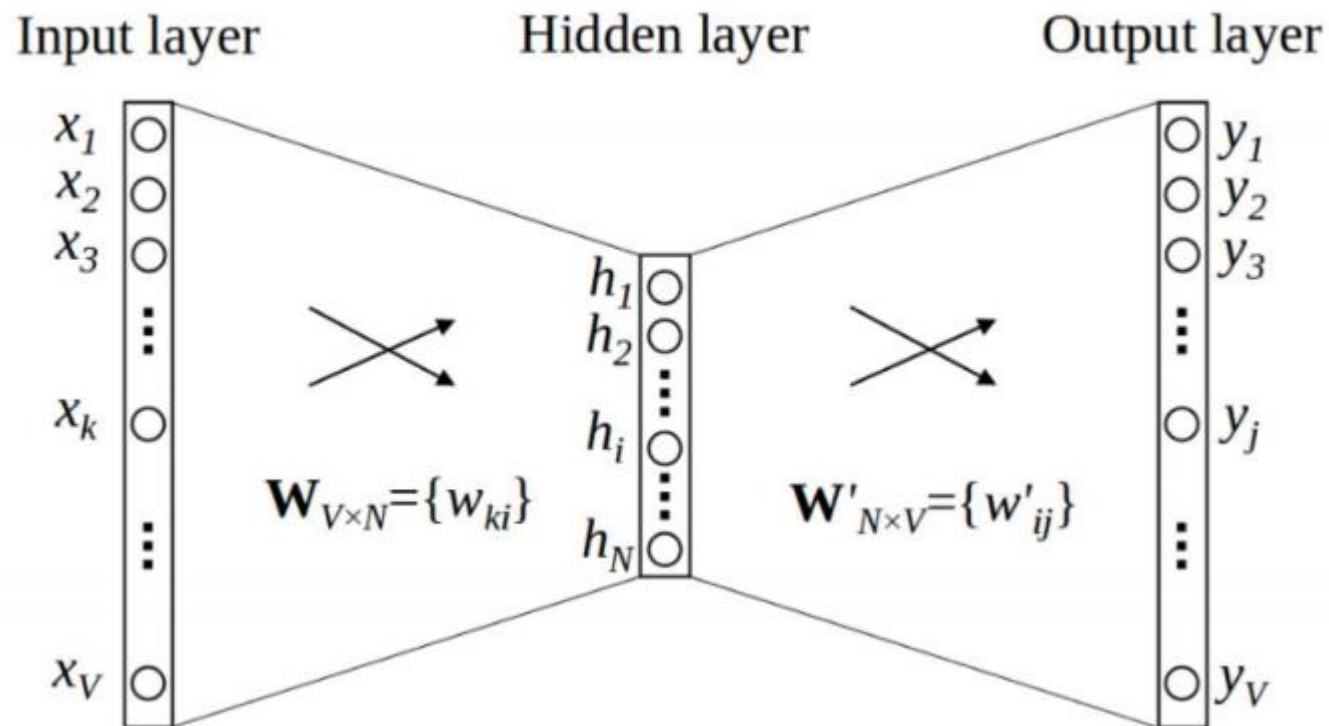


- В 2013 г. Томас Миколов и его коллеги предложили word2vec — упрощенную нейронную сеть, которую можно быстро обучить на огромном объёме текстов для получения векторов слов.

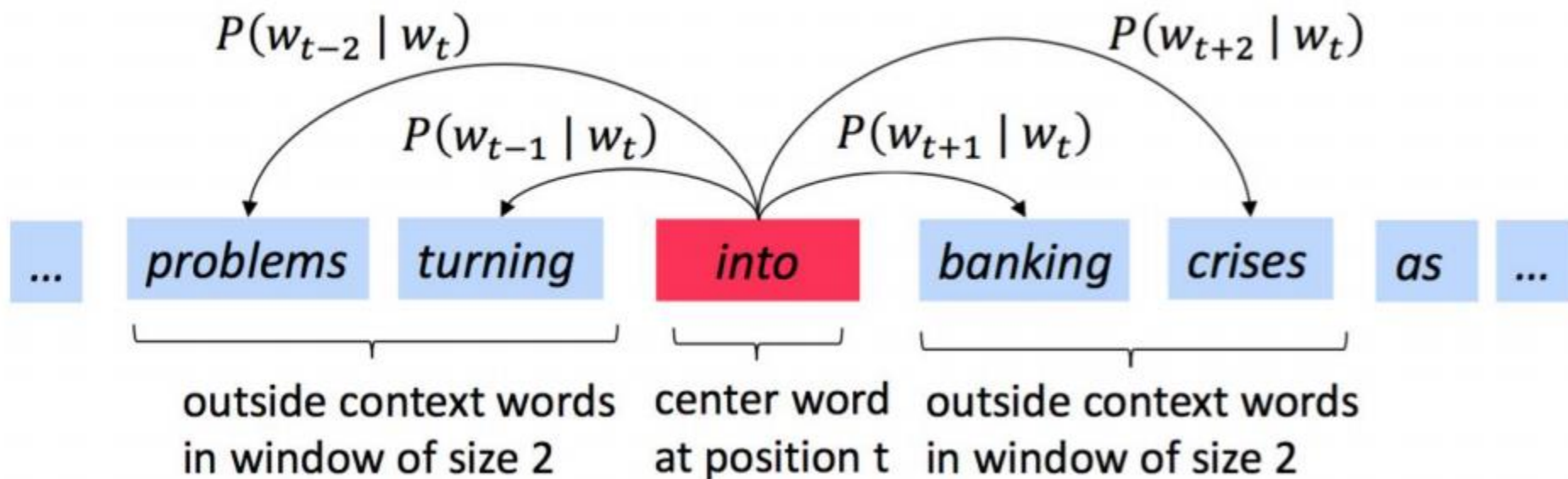


- T. Mikolov, K. Chen, G. Corrado, J. Dean. [Efficient Estimation of Word Representations in Vector Space](#) (2013).
- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. [Distributed Representations of Words and Phrases and their Compositionality](#) (2013).

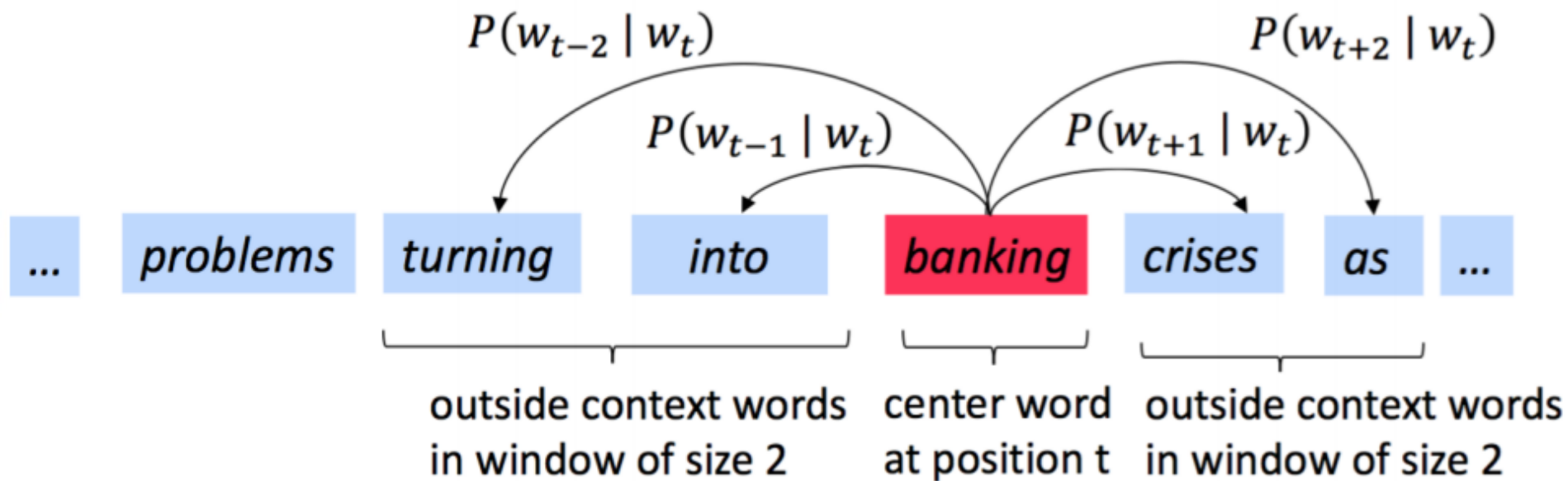
- Большой корпус текста
- Каждое слово в фиксированном словаре представлено вектором



- Пример окон и процесса для вычисления $P(w_{t+j} | w_t)$.



- Пример окон и процесса для вычисления $P(w_{t+j} | w_t)$.



word2vec: objective function

- Для каждой позиции $t = 1, \dots, T$ предсказываем слова контекста в окне фиксированного размер m , для заданного центрального слова w_t .
- Максимизируем функцию

$$L(\theta) = \prod_{t=1}^T \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} \mid w_t; \theta)$$

word2vec: objective function

- Для каждой позиции $t = 1, \dots, T$ предсказываем слова контекста в окне фиксированного размер m , для заданного центрального слова w_t .
- Максимизируем функцию

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^T \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} | w_t; \theta)$$

*θ is all variables
to be optimized*

word2vec: objective function

- **objective function (or loss, or cost function)** $J(\theta)$ это усредненный отрицательный логарифм правдоподобия

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} \mid w_t; \theta)$$

word2vec: objective function

- **objective function (or loss, or cost function)** $J(\theta)$ это усредненный отрицательный логарифм правдоподобия

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} \mid w_t; \theta)$$

word2vec: objective function

- **objective function (or loss, or cost function)** $J(\theta)$ это усредненный отрицательный логарифм правдоподобия

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$

Minimizing objective function  Maximizing predictive accuracy

word2vec: objective function

- Минимизируем функцию:
$$J(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$
- Вопрос:** как вычислить $P(w_{t+j} | w_t, \theta)$?
- Ответ:** используем два вектора для слова w :
 - Вектор для центрального слова v_w
 - Вектор контекста u_w
- Для центрального слова c и контекста слова o :

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

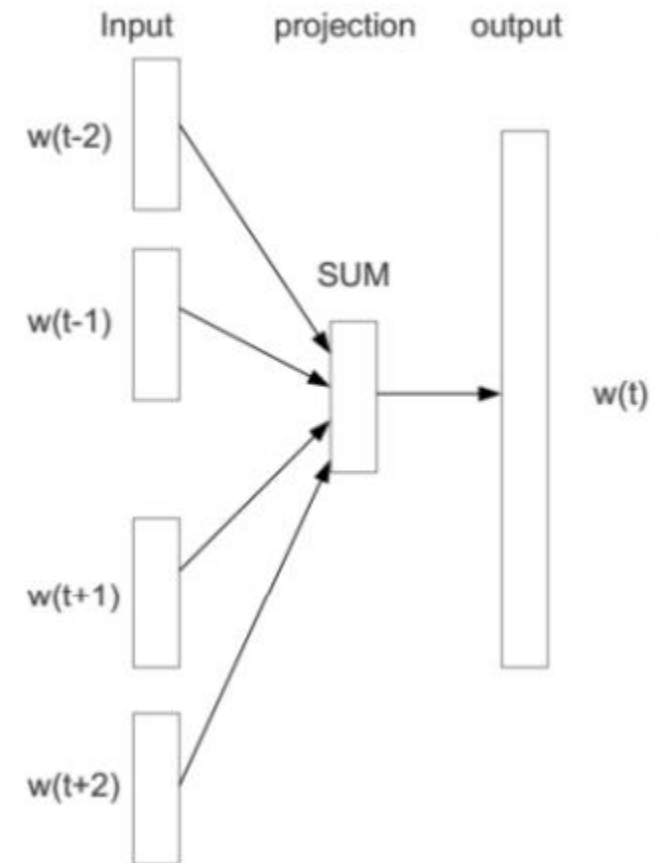
word2vec: параметры

- θ - d -размерные вектора для V слов;
- Каждое слово имеет два вектора;
- Мы оптимизируем эти параметры!

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \vdots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \vdots \\ u_{zebra} \end{bmatrix} \in \mathbb{R}^{2dV}$$

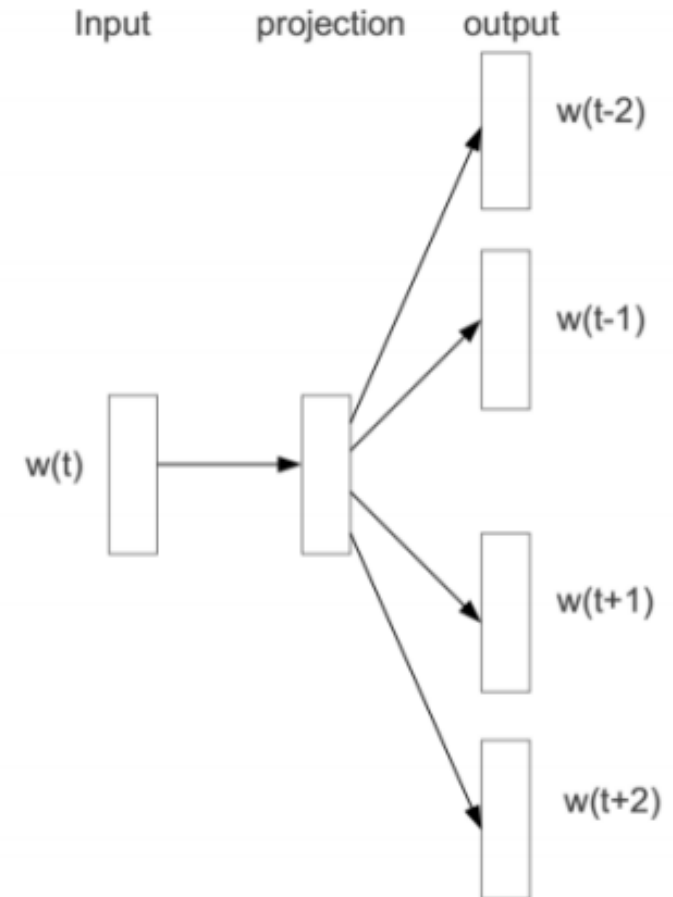
Continuous bag-of-words model (CBOW) [MCCD13]

- **Задача:** предсказание слова по заданному контексту.
- **Входной слой:**
 - контекст слова ($+$, $-\frac{k}{2}$ слова слева и справа)
- **Слой проекции:**
 - линейный
- **Выходной слой:**
 - вектор слова




skip-gram [MCCD13]


- **Обратная задача:** предсказание векторов контекста по данному слову
- **Выходной слой:**
 - вектор слов
- **Все контексты независимы:**
 - $(w, c_1), \dots, (w, c_k)$



Большая сумма, время для вычисления градиентов пропорционально $|V|$.


$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$


Большая сумма, время для вычисления градиентов пропорционально $|V|$.

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$


- Возможные решения:
 - Hierarchical softmax
 - Negative sampling

Большая сумма, время для вычисления градиентов пропорционально $|V|$.

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$


- Возможные решения:
 - Hierarchical softmax
 - Negative sampling

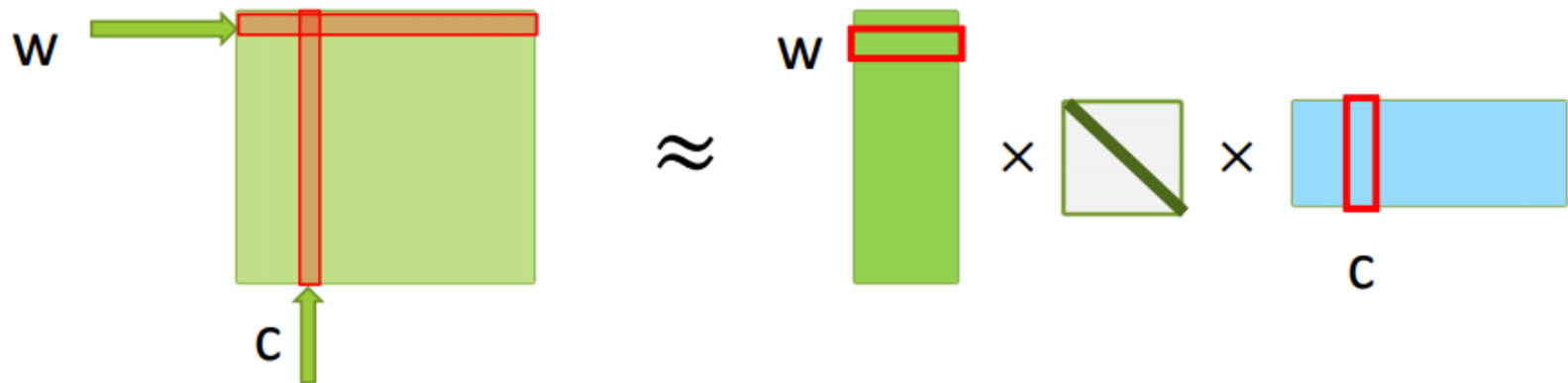
$$\sum_{w \in V} \exp(u_w^T v_c) \longrightarrow \sum_{w \in \{o\} \cup S_k} \exp(u_w^T v_c)$$

- Суммируем по небольшому подмножеству: negative sample, $|S_k| = k$.

word2vec с negative sampling как разложение матрицы PMI

$$PMI(w, c) = \log \frac{N(w, c) \times |V|}{N(w)N(c)}$$

$$PMI = X \approx \hat{X} = V_d \Sigma_d U_d^T$$



- O. Levy, Y. Goldberg. Neural Word Embedding as Implicit Matrix Factorization (2014).
- Minh Ngoc Le. <https://minhlab.wordpress.com/2015/06/> (2015).

Линейные свойства

- Оказывается, линейные операции над векторами v_w соответствуют семантическим преобразованиям!

$$V_{king} - V_{man} + V_{woman} \approx V_{queen}.$$

$$V_{Paris} - V_{France} + V_{Italy} \approx V_{Rome}.$$

$$V_{big} - V_{small} + V_{smallest} \approx V_{biggest}.$$

Многообразие эмбедингов

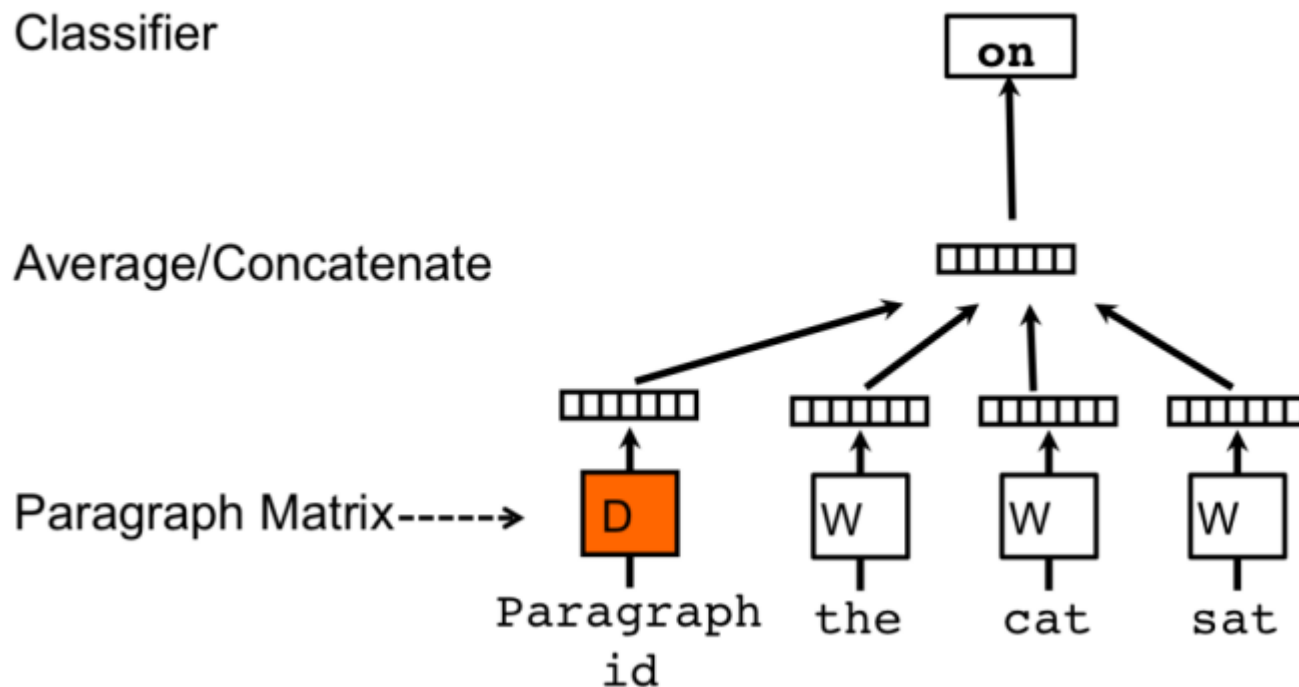
- Skip-gram, CBOW aka word2vec [Mikolov et al., 2013]
 - <http://rusvectors.org/ru/>
 - <https://www.tensorflow.org/tutorials/word2vec>
- Dependency embeddings [Levi et al., 2015]
 - <https://bitbucket.org/yoavgo/word2vecf>
- GloVe [Pennington et al., 2014]
 - <https://nlp.stanford.edu/projects/glove/>
- FastText [Joulin et al., 2016]
 - <https://github.com/facebookresearch/fastText>
- AdaGram [Bartunov et al., 2016]
 - <https://github.com/sbos/AdaGram.jl>
 - <http://adagram.ll-cl.org>
- SenseGram [Peleвина et al., 2016]
 - <https://github.com/tudarmstadt-lt/sensegram>
- StarSpace [Wu, 2017]
 - <https://github.com/facebookresearch/StarSpace>
- Poincare embeddings [Nickel et al., 2017]

Сравнение моделей эмбедингов [SLMJ15]

- **Внутренние (intrinsic) задачи**
 - Определение похожих слов
 - Определение аналогий
 - Определение объектов глаголов
- **Внешние (extrinsic) задачи**
 - Классификация текстов
 - Извлечение именованных сущностей
 - Расширение запроса
- **Результаты** зависят от использованного корпуса для обучения, гиперпараметров обучения, корпуса для тестирования. Невозможно определить модель эмбедингов, превосходящую остальные.

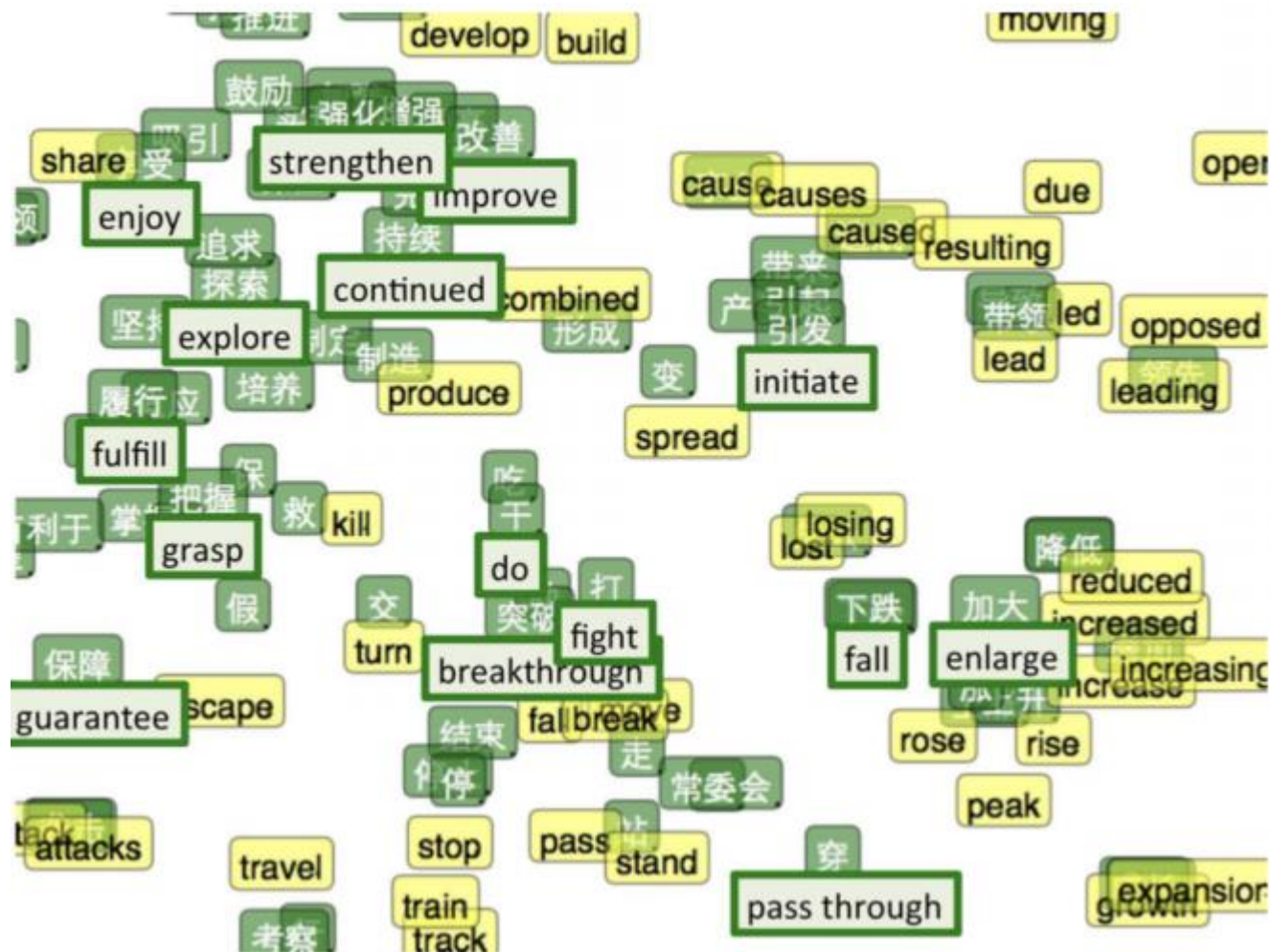
Насколько похожи два предложения (абзаца)? [LM14]

- Как найти вектор-предложения (абзаца)?
- Усреднить вектора слов, входящих в каждое предложение (с tf – idf весами)
- Doc2vec: что word2vec, только для предложений (абзацев).



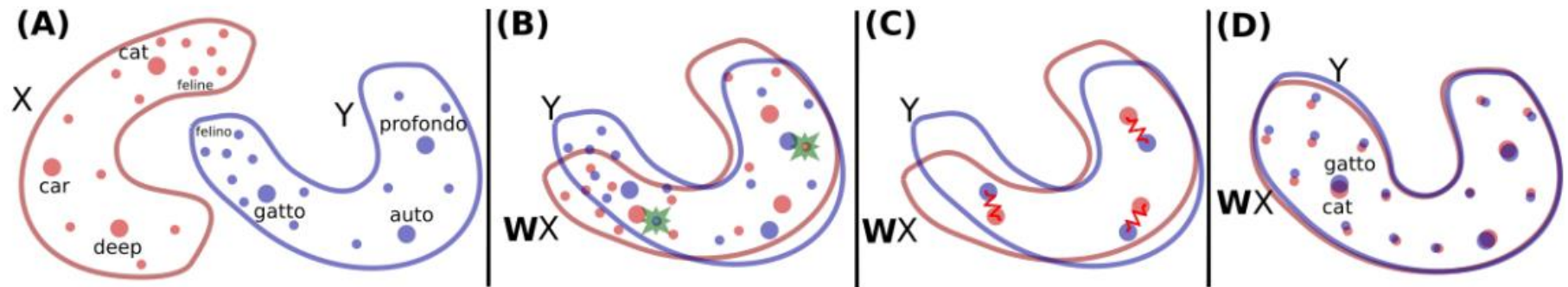
Двуязычные эмбеддинги [ZSCM13]

- Дан (выровненный) параллельный корпус. Контекст слова: перевод этого слова на другой язык.



Двуязычные эмбединги [CLR+17]

- Дано два невыровненных пространства слов
- Adversarial learning для определения матрицы поворота W
- Прокрустово преобразование для уточнения W
- k – NN-подобный метод для окончательного выравнивания



Когда нам учить эмбединги?

СПАСИБО ЗА ВНИМАНИЕ

Литература

- NLP курс в яндексе https://github.com/yandexdataschool/nlp_course
- Курс в Stanford CS224N <http://cs224n.stanford.edu/>

- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, [Enriching word vectors with subword information](#), arXiv preprint arXiv:1607.04606 (2016).
- Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov, [Breaking sticks and ambiguities with adaptive skip-gram](#), Artificial Intelligence and Statistics, 2016, pp. 130–138.
- Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, [Word translation without parallel data](#), arXiv preprint arXiv:1710.04087 (2017).
- William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky, [Inducing domain-specific sentiment lexicons from unlabeled corpora](#), Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing, vol. 2016, NIH Public Access, 2016, p. 595.
- William L Hamilton, Jure Leskovec, and Dan Jurafsky, [Diachronic word embeddings reveal statistical laws of semantic change](#), arXiv preprint arXiv:1605.09096 (2016).

- Andrey Kutuzov and Elizaveta Kuzmenko, [Webvectors: a toolkit for building web interfaces for vector semantic models](#), International Conference on Analysis of Images, Social Networks and Texts, Springer, 2016, pp. 155–16
- Omer Levy and Yoav Goldberg, [Dependency-based word embeddings](#), Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, 2014, pp. 302–308. 1.
- Quoc Le and Tomas Mikolov, [Distributed representations of sentences and documents](#), International Conference on Machine Learning, 2014, pp. 1188–1196.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, [Efficient estimation of word representations in vector space](#), arXiv preprint arXiv:1301.3781 (2013).
- Jeffrey Pennington, Richard Socher, and Christopher Manning, [Glove: Global vectors for word representation](#), Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

- Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims, [Evaluation methods for unsupervised word embeddings](#), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 298–307.
- Joseph Turian, Lev Ratinov, and Yoshua Bengio, [Word representations: a simple and general method for semi-supervised learning](#), Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics, 2010, pp. 384–394.
- Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning, [Bilingual word embeddings for phrase-based machine translation](#), Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1393–1398.