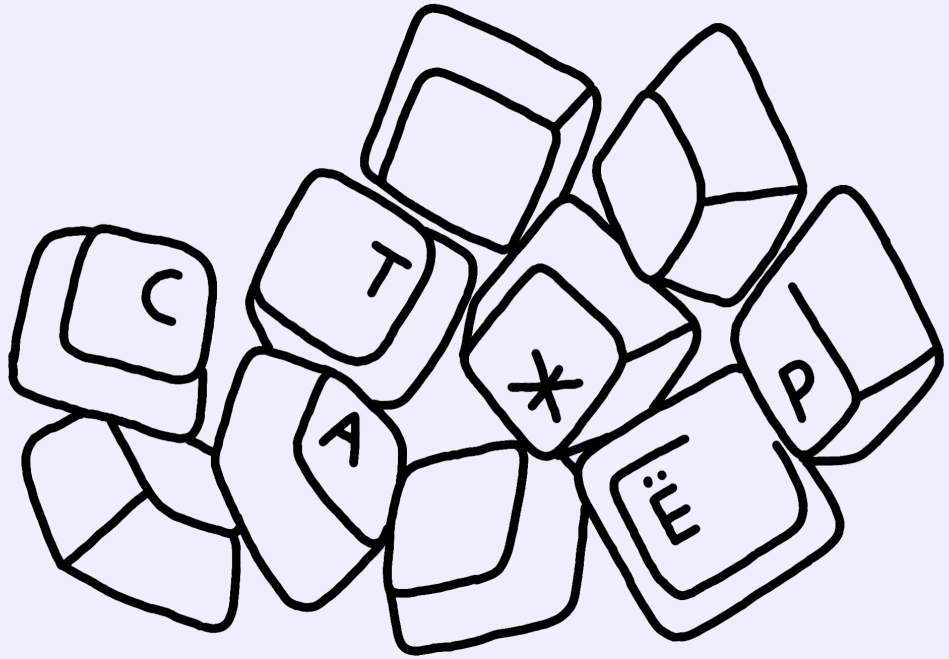
Информативные векторные представления в машинном обучении

YOUNG & YANDEX

Радослав Нейчев
Выпускник и преподаватель ШАД и МФТИ,
руководитель группы ML-разработки в Яндексе,
сооснователь girafe-ai

Содержание

Правдоподобие / Likelihood



01

Правдоподобие / Likelihood

Пусть задана выборка X, Y и модель с параметрами θ.

Правдоподобием будем называть

$$L(\theta|X,Y) = P(X,Y|\theta) \to \max_{\theta}$$

помним про i.i.d.

$$P(X,Y|\theta) = \prod_i P(x_i, y_i|\theta)$$

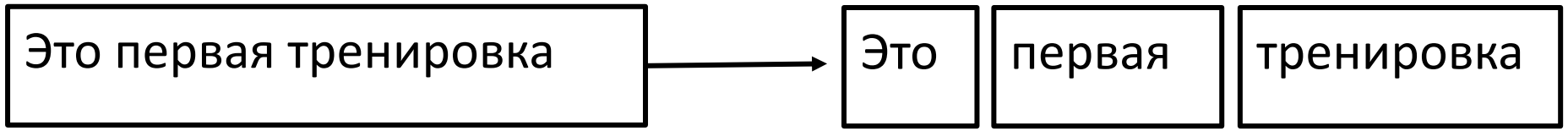
$$\log L(\theta|X,Y) = \log P(X,Y|\theta) = \sum_i \log P(x_i, y_i|\theta)$$

Векторные представления для текста

02

Токенизация

Токен – атомарный элемент последовательности.



Токеном может быть как слово, так и символ, и морфема – это вопрос договоренности в каждой задаче.

Мешок слов – Bag-of-Words

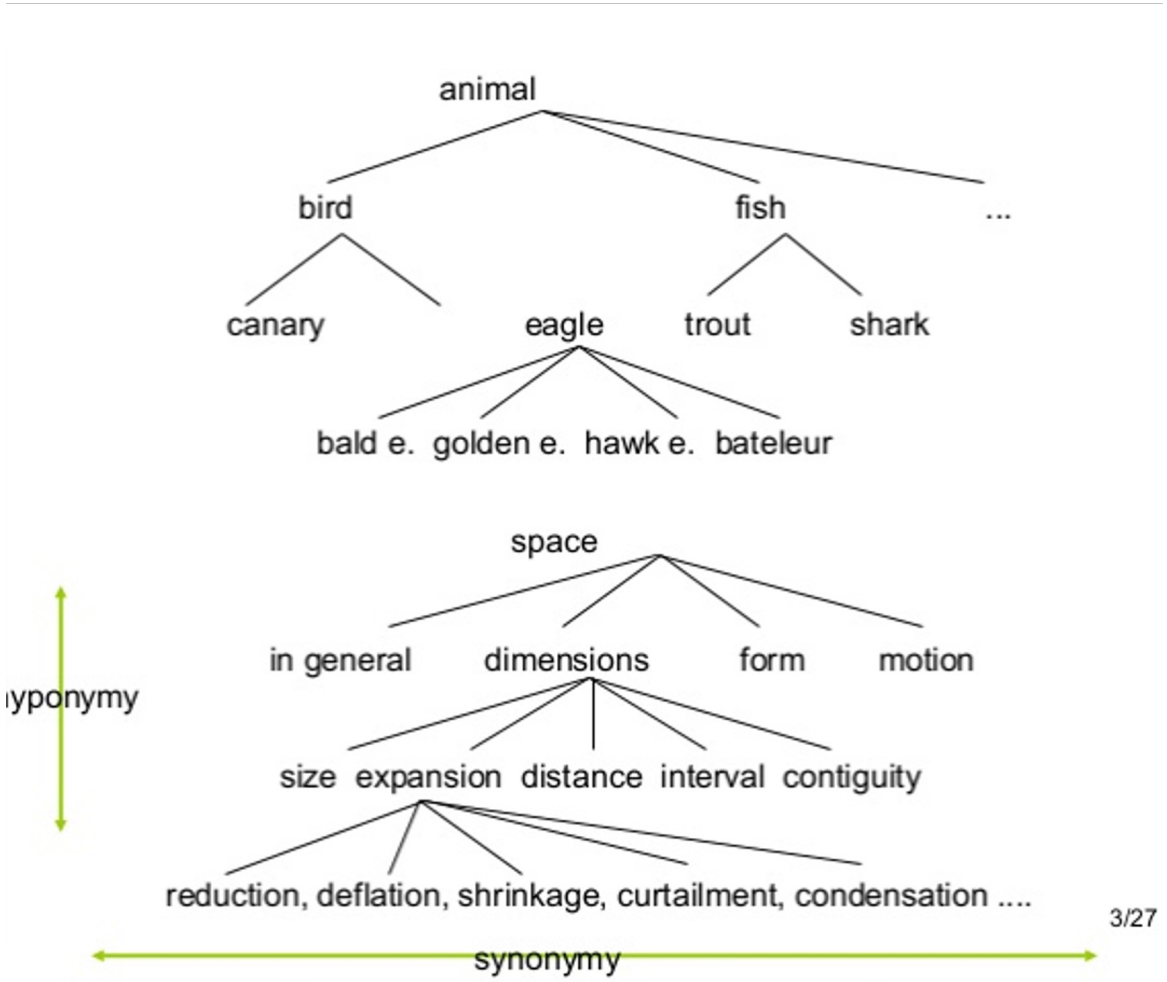
Как-то раз купил мужик шляпу – а она ему как раз

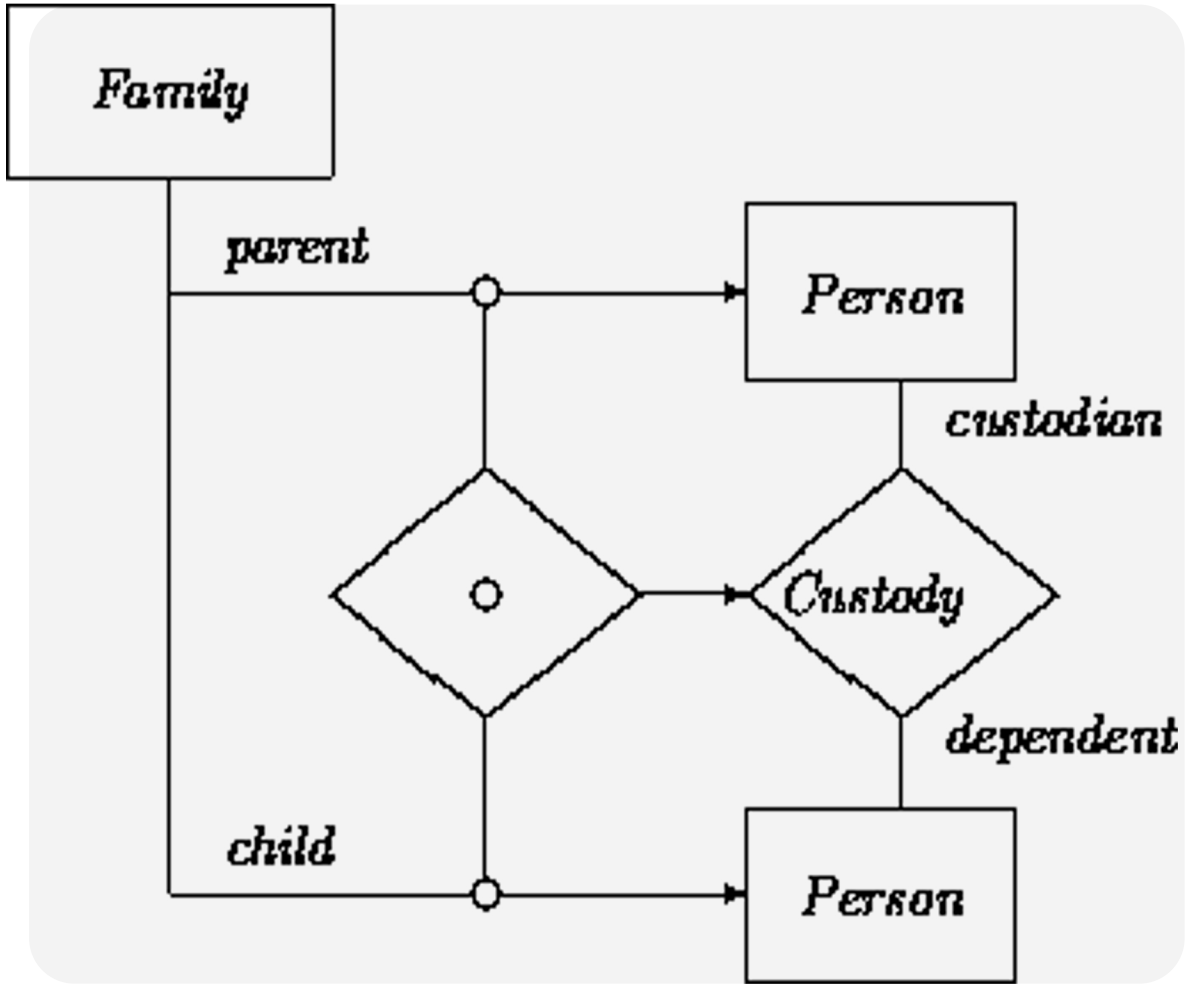
день	как	то	раз	купил	мужик	шляпу	а	она	ему	кот	гав
0	2	1	2	1	1	1	1	1	1	0	0

Минусы:

- Теряем информацию о порядке слов
- Векторы высокой размерности
- Векторы крайне разреженные
- Разные формы слов воспринимаются как разные слова

WordNet





Read more: https://en.wikipedia.org/wiki/WordNet

Чем предобрабатывать тексты?

- NLTK
  - nltk.stem.SnowballStemmer
  - nltk.stem.PorterStemmer
  - nltk.stem.WordNetLemmatizer
  - nltk.corpus.stopwords
- BeautifulSoup (for parsing HTML)
- Regular Expressions (import re)
- Pymorphy2

TF-IDF

- Term Frequency (tf): gives us the frequency of the word in each document in the corpus.

$$\mathrm{tf}(t,d) = f_{t,d}$$

- Inverse Document Frequency (idf): used to calculate the weight of rare words across all documents in the corpus. The words that occur rarely in the corpus have a high IDF score.

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

N: total number of documents in the corpus $N = |D|$

$|\{d \in D : t \in d\}|$: number of documents where the term t

TF-IDF

- *Sentence A:* The car is driven on the road.
- *Sentence B:* The truck is driven on the highway.

(each sentence is a separate document)

TF-IDF

Word	TF		IDF	TF * IDF	
	A	B		A	B
The	1/7	1/7			
Car	1/7	0			
Truck	0	1/7			
Is	1/7	1/7			
Driven	1/7	1/7			
On	1/7	1/7			
The	1/7	1/7			
Road	1/7	0			
Highway	0	1/7			

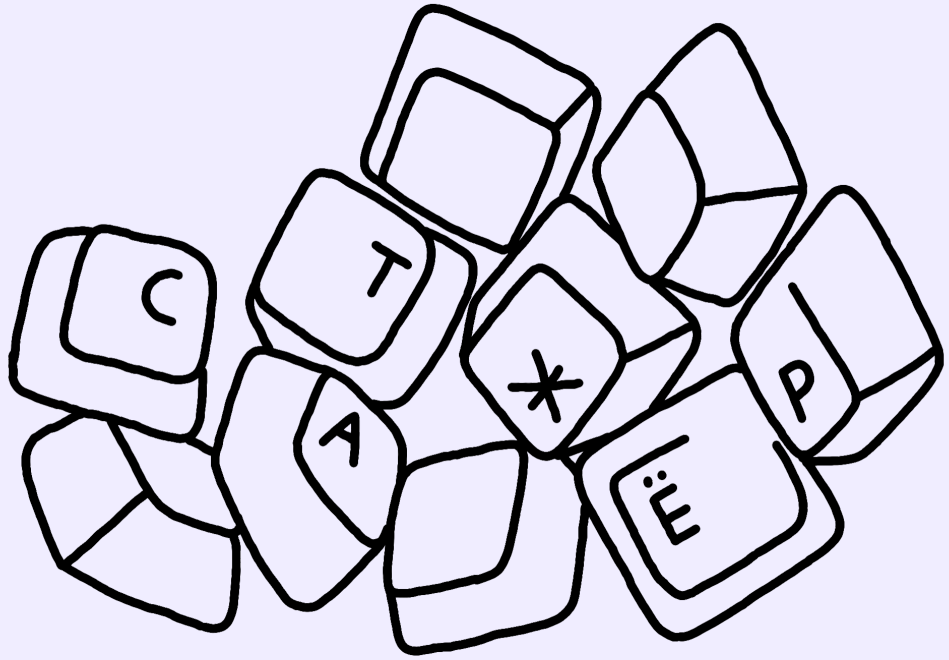
TF-IDF

Word	TF		IDF	TF * IDF	
	A	B		A	B
The	1/7	1/7	log(2/2)=0		
Car	1/7	0	log(2/1)=0.3		
Truck	0	1/7	log(2/1)=0.3		
Is	1/7	1/7	log(2/2)=0		
Driven	1/7	1/7	log(2/2)=0		
On	1/7	1/7	log(2/2)=0		
The	1/7	1/7	log(2/2)=0		
Road	1/7	0	log(2/1)=0.3		
Highway	0	1/7	log(2/1)=0.3		

TF-IDF

Word	TF		IDF	TF * IDF	
	A	B		A	B
The	1/7	1/7	log(2/2)=0	0	0
Car	1/7	0	log(2/1)=0.3	0.043	0
Truck	0	1/7	log(2/1)=0.3	0	0.043
Is	1/7	1/7	log(2/2)=0	0	0
Driven	1/7	1/7	log(2/2)=0	0	0
On	1/7	1/7	log(2/2)=0	0	0
The	1/7	1/7	log(2/2)=0	0	0
Road	1/7	0	log(2/1)=0.3	0.043	0
Highway	0	1/7	log(2/1)=0.3	0	0.043

Информативные векторные представления – эмбеддинги



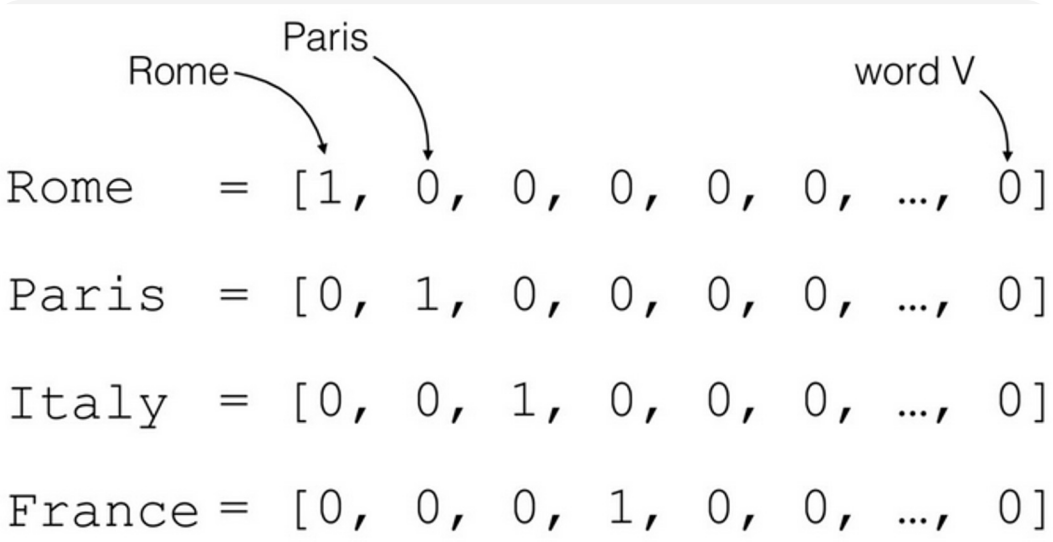
03

Простейший вариант – one-hot

Проблемы с one-hot:

- Высокая размерность
- Разреженность
- Все векторы взаимно ортогональны

One-hot vectors:



Как определить значение слов, если нельзя пользоваться словами?

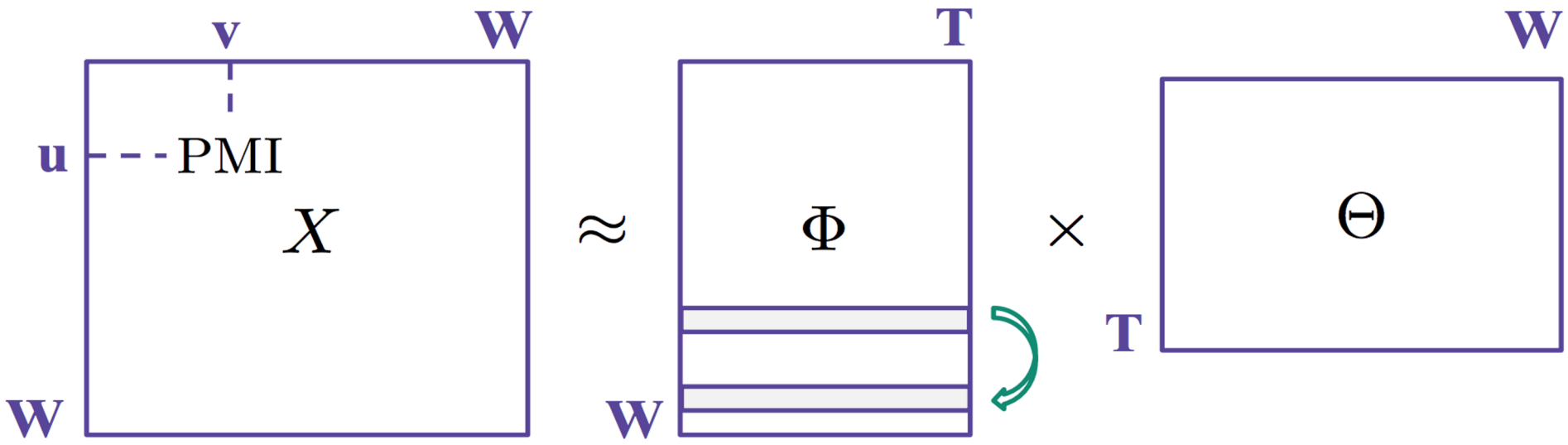
“You shall know a word by the company it keeps”

J. R. Firth, 1957: 11

…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*

source: https://web.stanford.edu/class/cs224n/slides/cs224n-spr2024-lecture01-wordvecs1.pdf

Вспомним матричные разложения

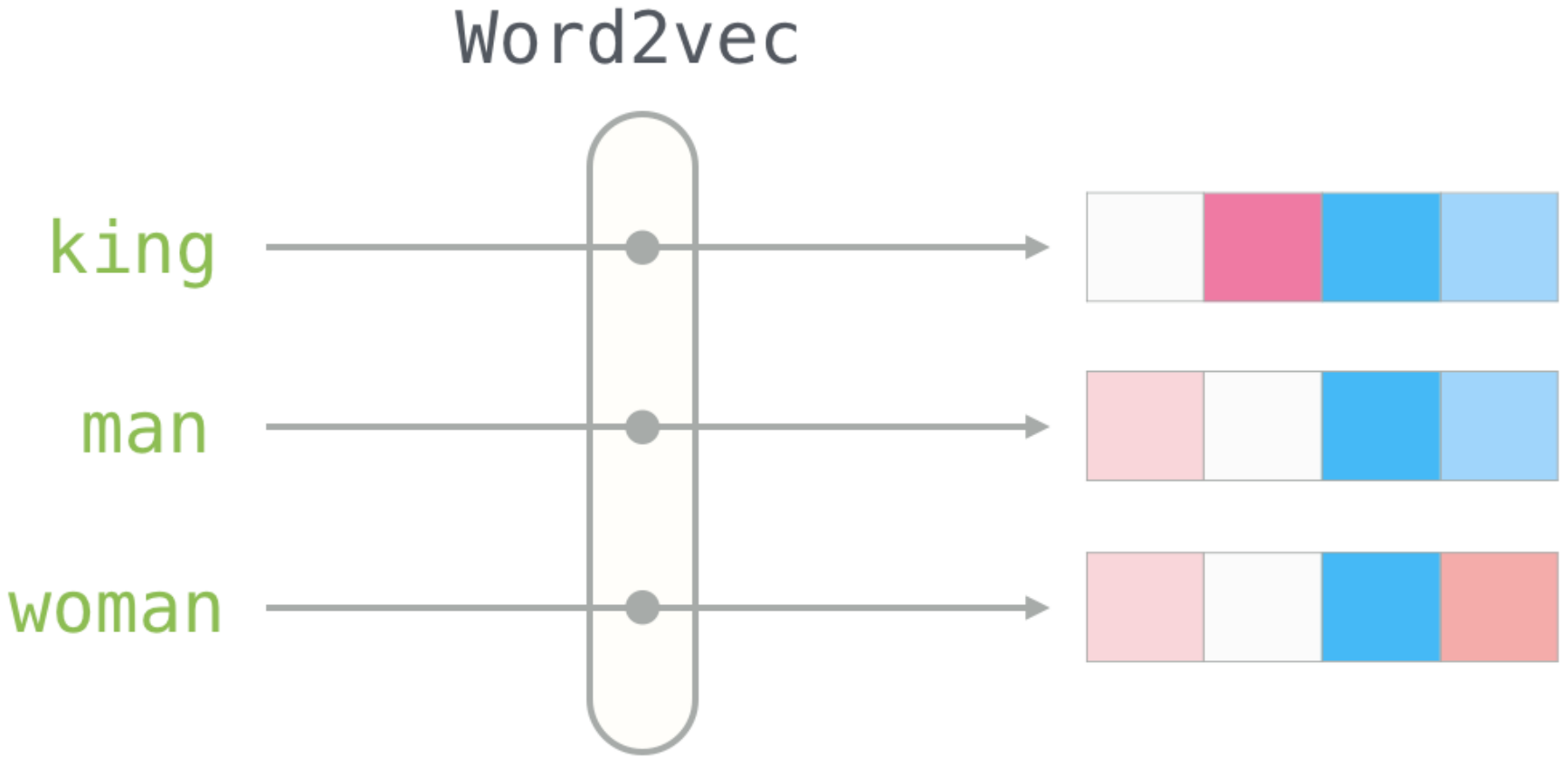


Построение эмбеддингов для слов word2vec

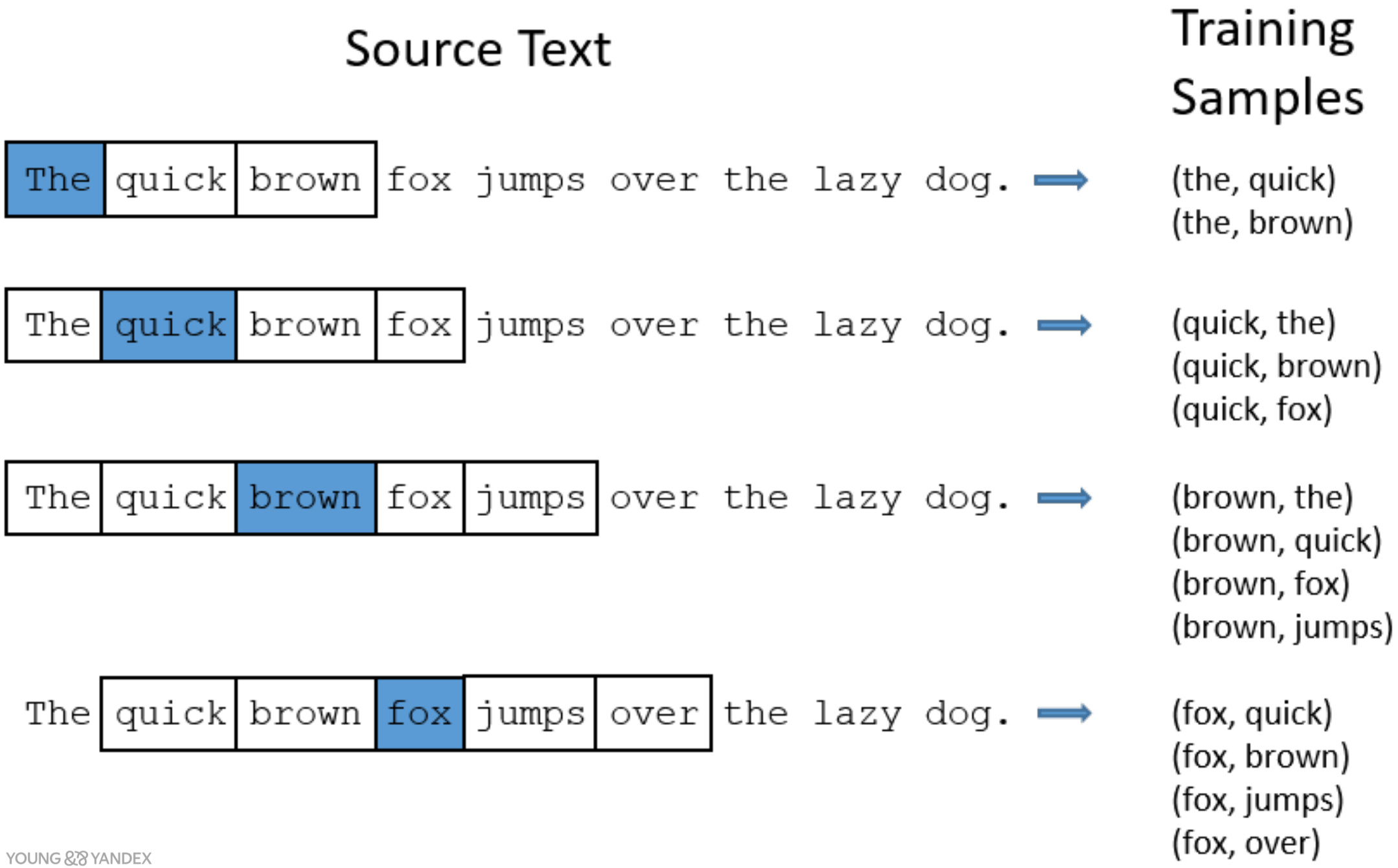
04

word2vec

word2vec – метод построения информативных векторных представлений слов, представлен в работе 2013 за авторством Thomas Mikolov и его коллег

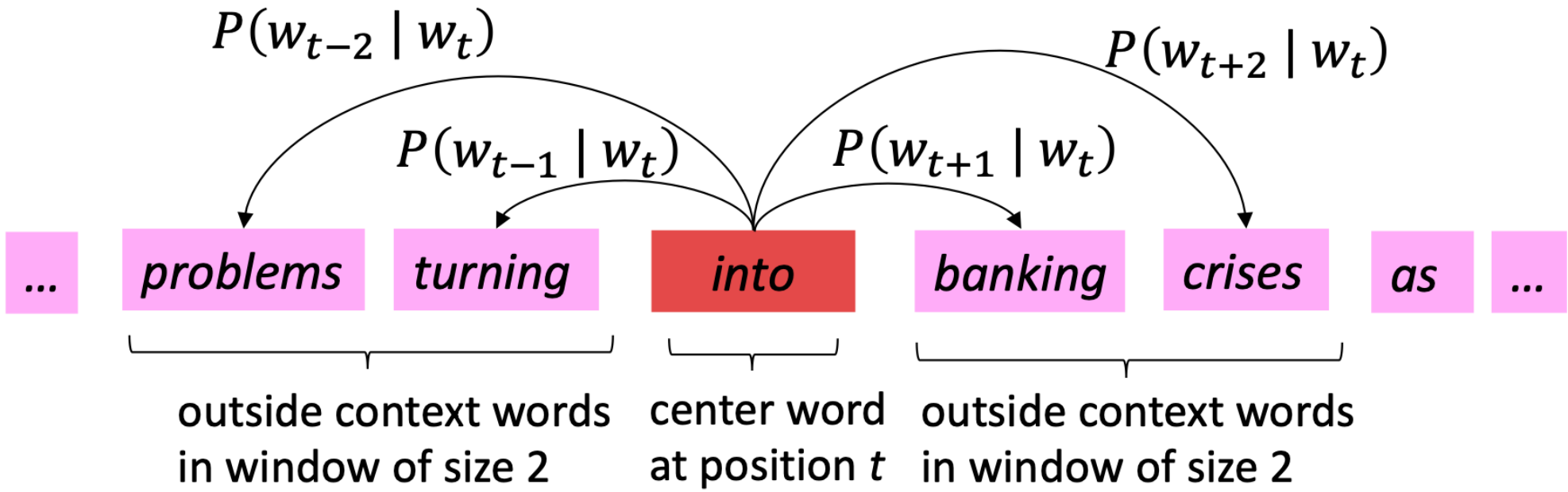


source: https://jalammar.github.io/illustrated-word2vec/



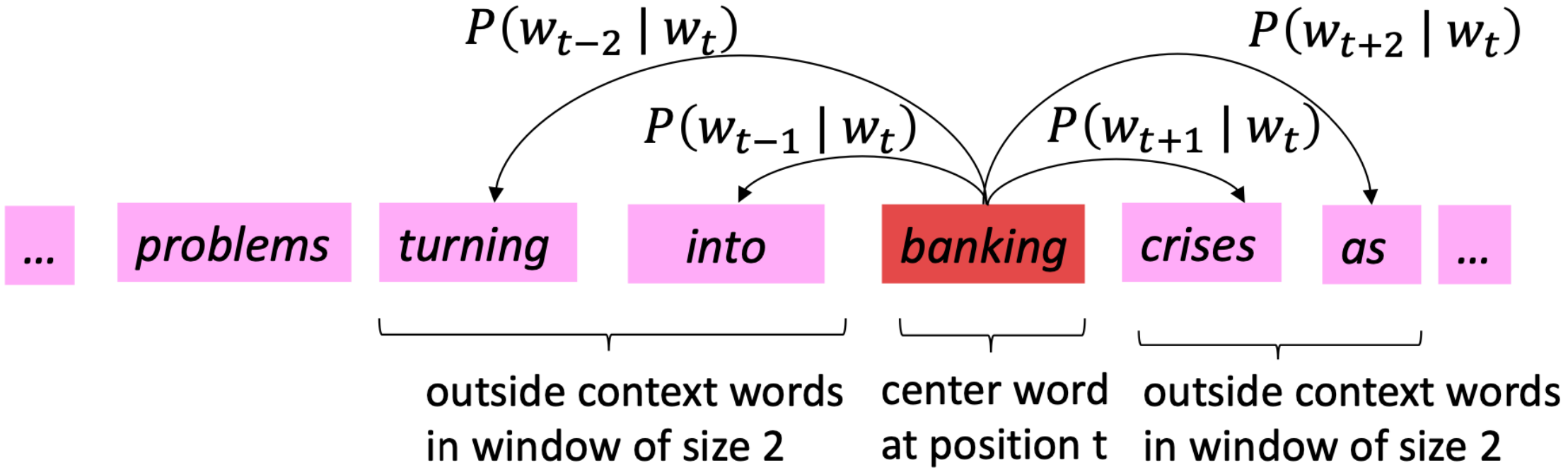
source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

word2vec



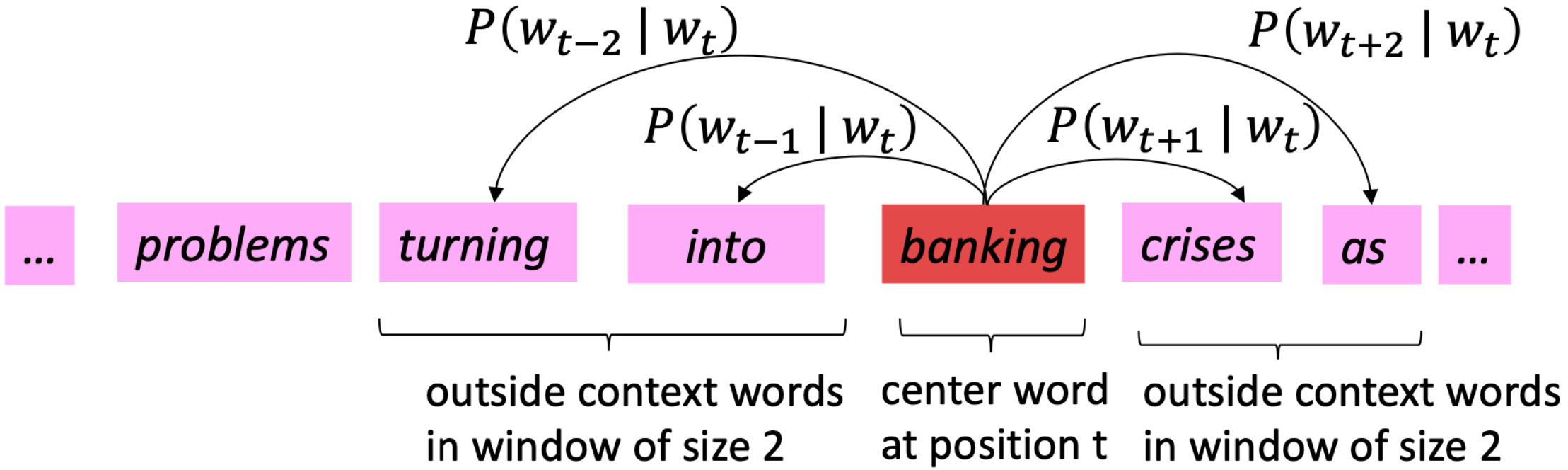
source: https://web.stanford.edu/class/cs224n/slides/cs224n-spr2024-lecture01-wordvecs1.pdf

word2vec



source: https://web.stanford.edu/class/cs224n/slides/cs224n-spr2024-lecture01-wordvecs1.pdf

word2vec



Максимизируемый функционал:
(логарифм правдоподобия)

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

source: https://web.stanford.edu/class/cs224n/slides/cs224n-spr2024-lecture01-wordvecs1.pdf

word2vec

Максимизируемый функционал:
(логарифм правдоподобия)

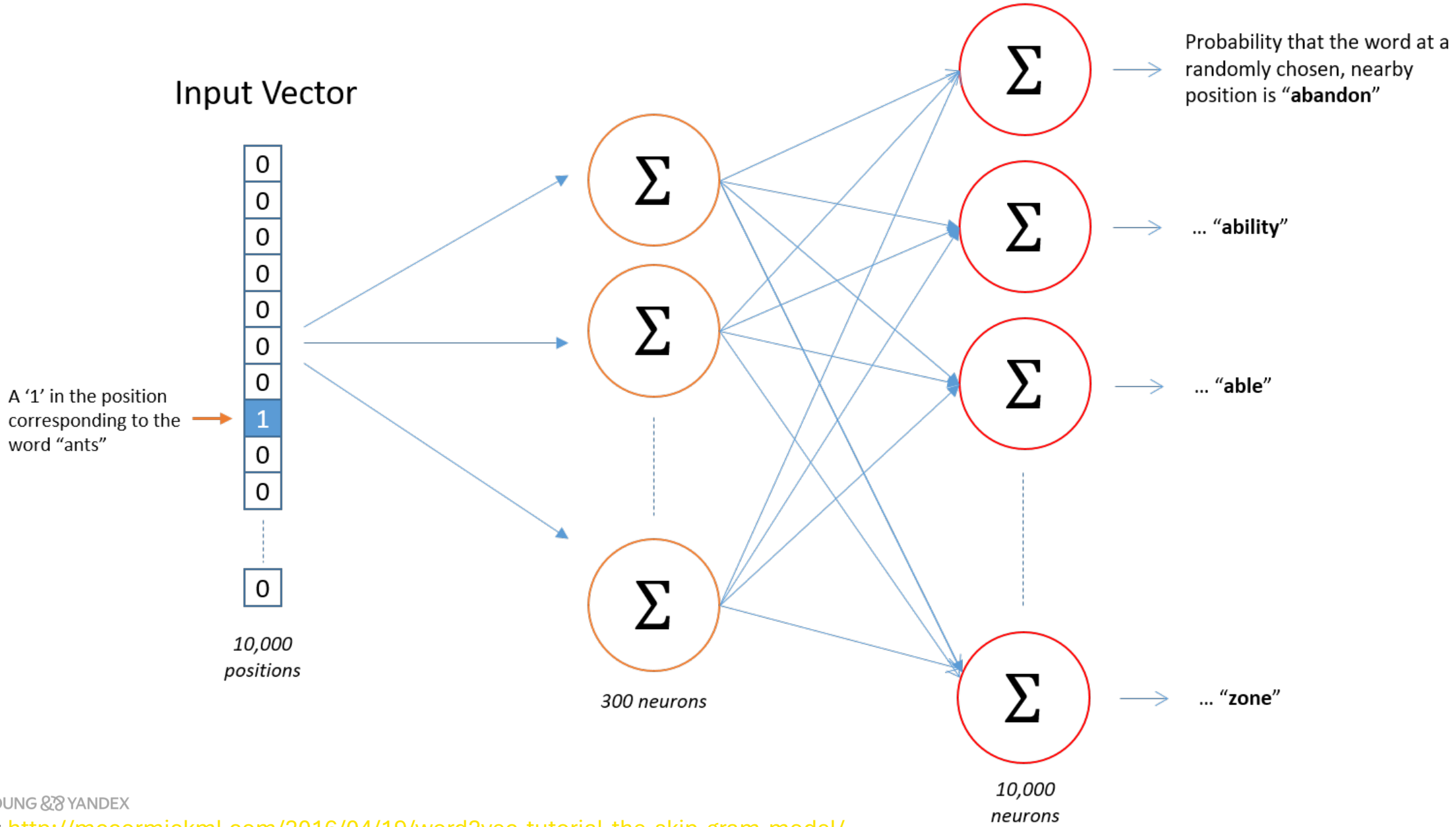
$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

Оценка вероятности через softmax:
(очевидный способ)

$$p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top}v_{w_I}\right)}$$

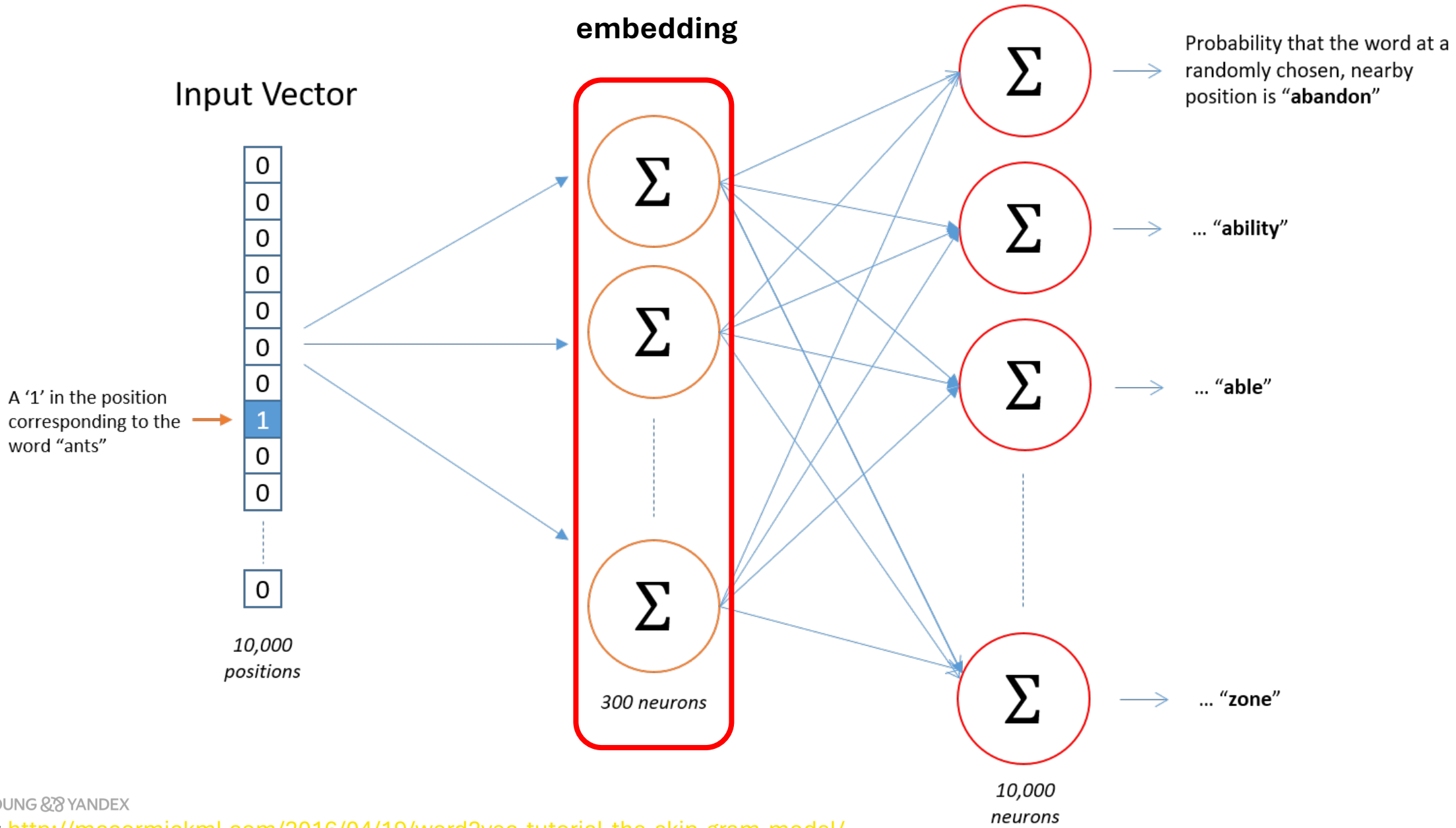
source: https://web.stanford.edu/class/cs224n/slides/cs224n-spr2024-lecture01-wordvecs1.pdf

word2vec

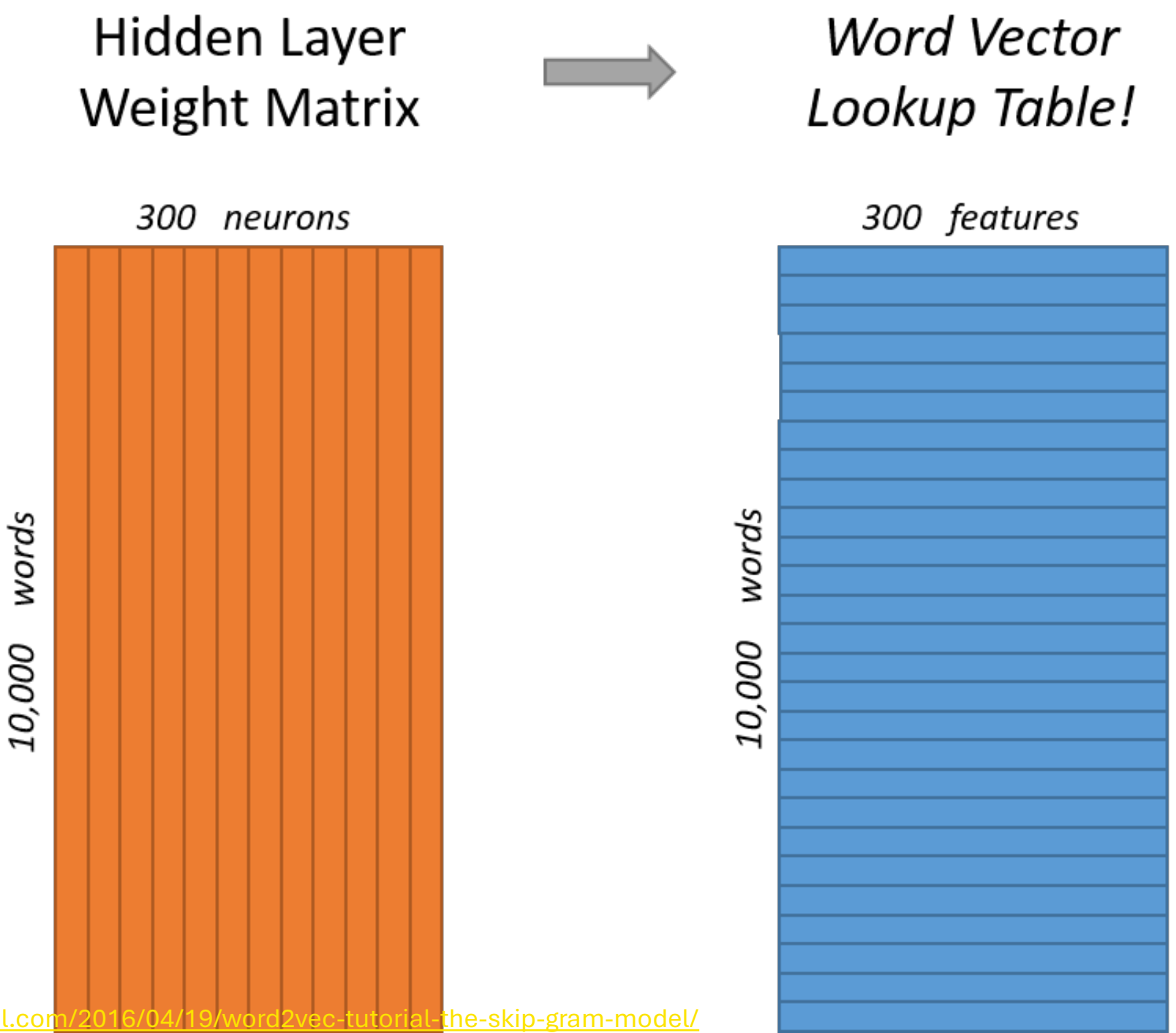


source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

word2vec



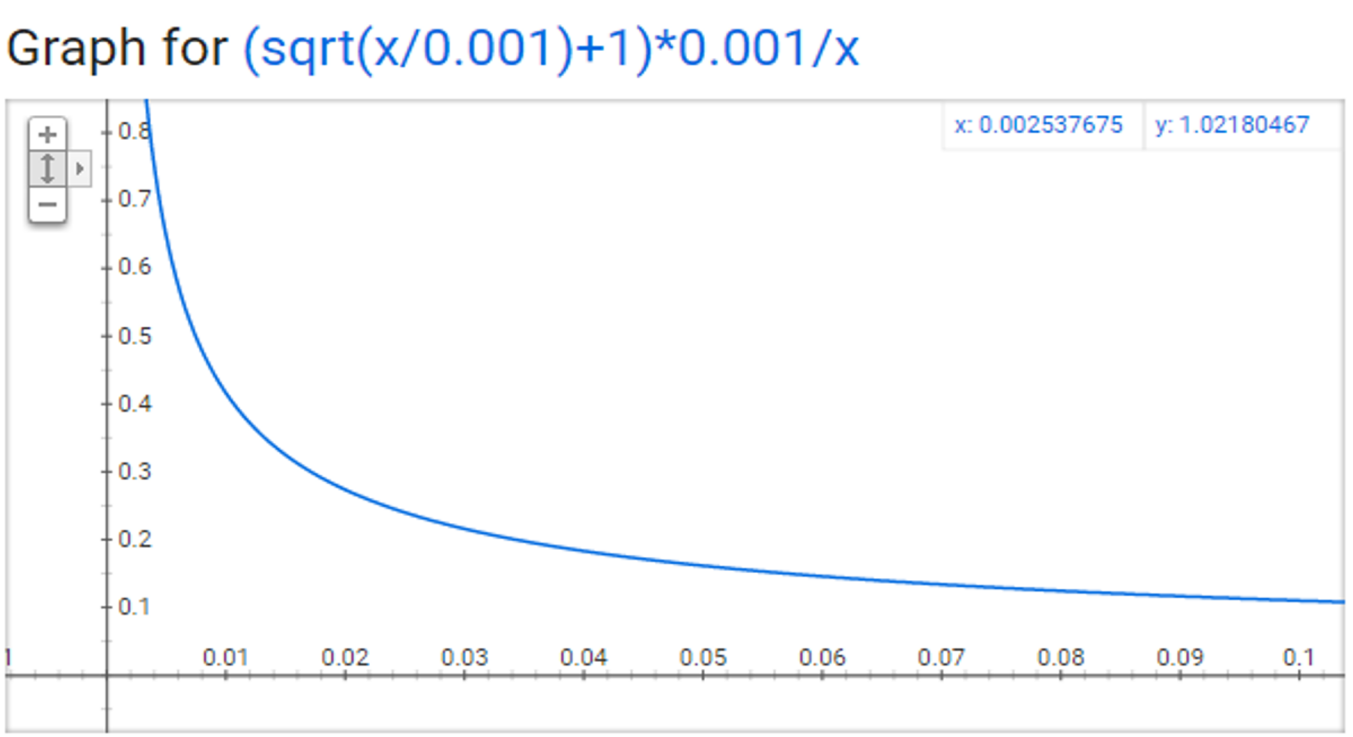
source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/



source: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

word2vec: subsampling

Часто встречающиеся слова чаще выступают в роли объектов – будем пропускать их случайным образом в зависимости от частоты слова $z(w_i)$



$P(w_i)$ – вероятность использовать пару в обучении

$$P(w_i) = (\sqrt{\frac{z(w_i)}{0.001}} + 1) \cdot \frac{0.001}{z(w_i)}$$

source: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

word2vec: negative sampling

Имеет смысл не только “сближать” похожие (близкие по контексту) слова, но и “отдалять” непохожие. Для этого воспользуемся механизмом negative sampling.

Чем чаще встречается слово в обучающем корпусе, тем больше вероятность использовать его в качестве negative sample.

$$P(w_i) = \frac{f(w_i)}{\sum_{j=0}^{n} \left(f(w_j)\right)}$$

source: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

word2vec: negative sampling

Имеет смысл не только “сближать” похожие (близкие по контексту) слова, но и “отдалять” непохожие. Для этого воспользуемся механизмом negative sampling.

Чем чаще встречается слово в обучающем корпусе, тем больше вероятность использовать его в качестве negative sample.

$$P(w_i) = \frac{f(w_i)}{\sum_{j=0}^{n}\left(f(w_j)\right)} \Longrightarrow P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n}\left(f(w_j)^{3/4}\right)}$$

source: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/

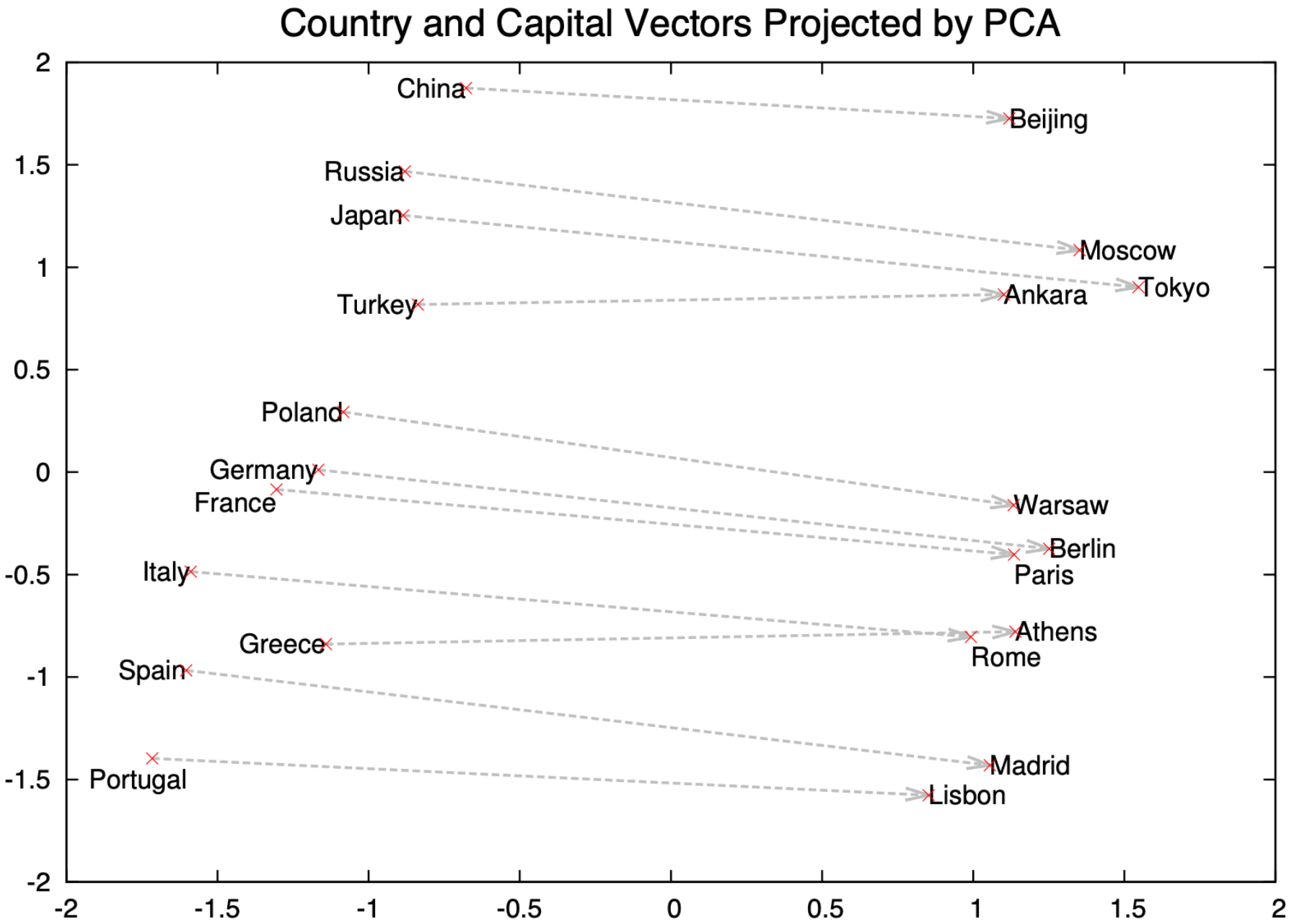
word2vec

Обновленный оптимизируемый функционал: рассматриваем лишь **положительный** пример и **несколько отрицательных:**

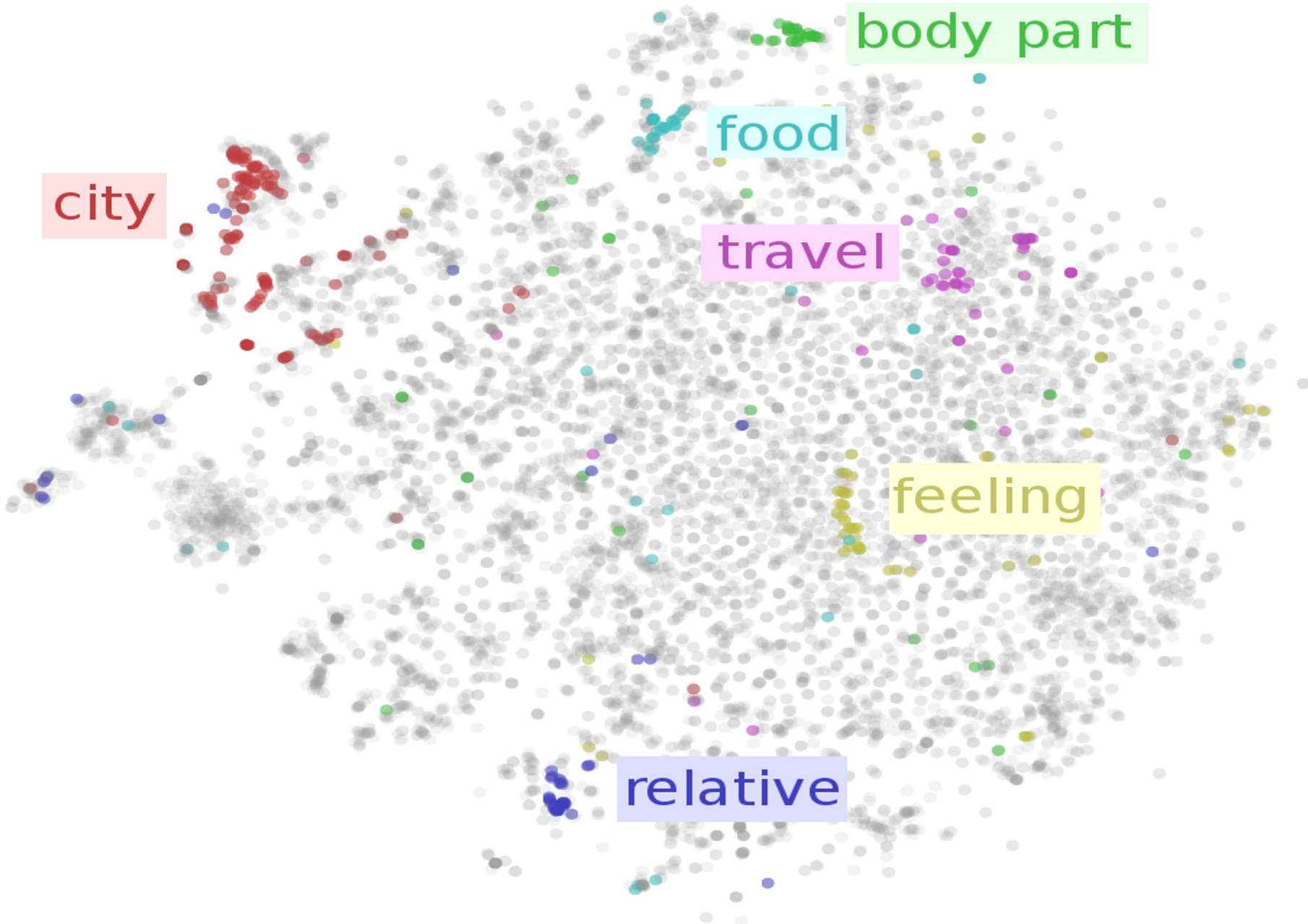
$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]$$

Перед нами один из многих примеров контрастного обучения – contrastive learning. В дальнейшем мы еще не раз столкнемся с ним.

source: Distributed Representations of Words and Phrases and their Compositionality



source: Distributed Representations of Words and Phrases and their Compositionality


body part
food
city
travel
feeling
relative

Спасибо за внимание

Y&Y