NLP
Word embeddings

Основано на [NLP Course | For You]

План

- Особенности домена
- Решаемые задачи
- Предобработка: токенизация и эмбединги

- One-hot
- Count based
- Word2Vec, FastText

Особенности текстового домена

- Слабая структурируемость
- На входе получаем не числа, а последовательность символов
- Длина текстов бывает разной
- Тексты сравнительно часто бывают с опечатками
- Сильная зависимость от контекста (например, кореференции)
- Грамматически и/или семантически связанные слова не всегда расположены рядом
- Неразмеченных данных очень много, даже с древних времён

Задачи решаемые в NLP

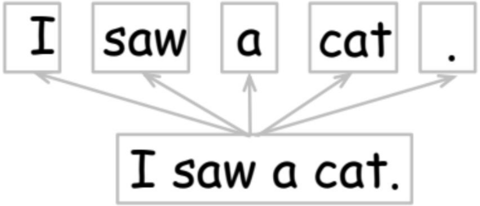
- Классификация текста
- Классификация слов
- Языковое моделирование (aka LM, language modeling)
- text-to-text
- Заполнение маскированных частей текста
- Схожесть текстов
- Выделение границ предложения, расстановка пунктуации

Токенизация

I saw a cat.

Text (your input)

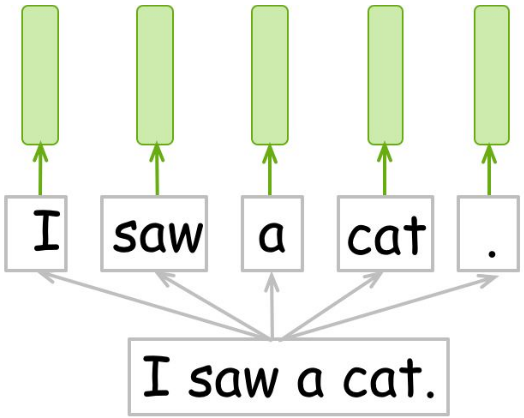
Токенизация



Sequence of tokens

Text (your input)

Embeddings

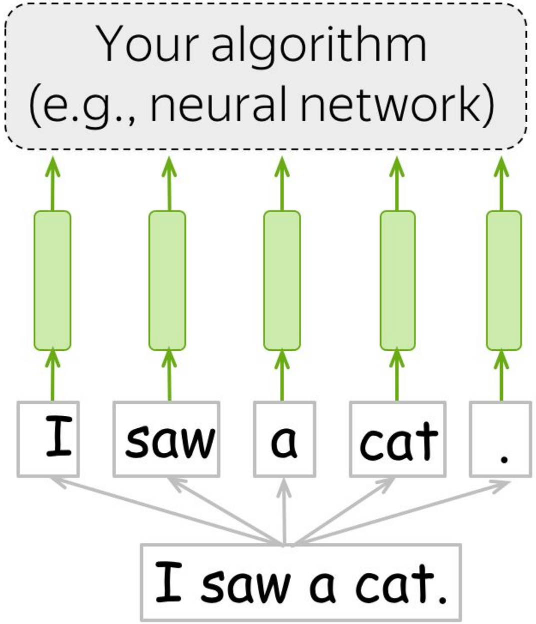


Word representation - vector
(input for your model/algorithm)

Sequence of tokens

Text (your input)

Обработка последовательности эмбедингов



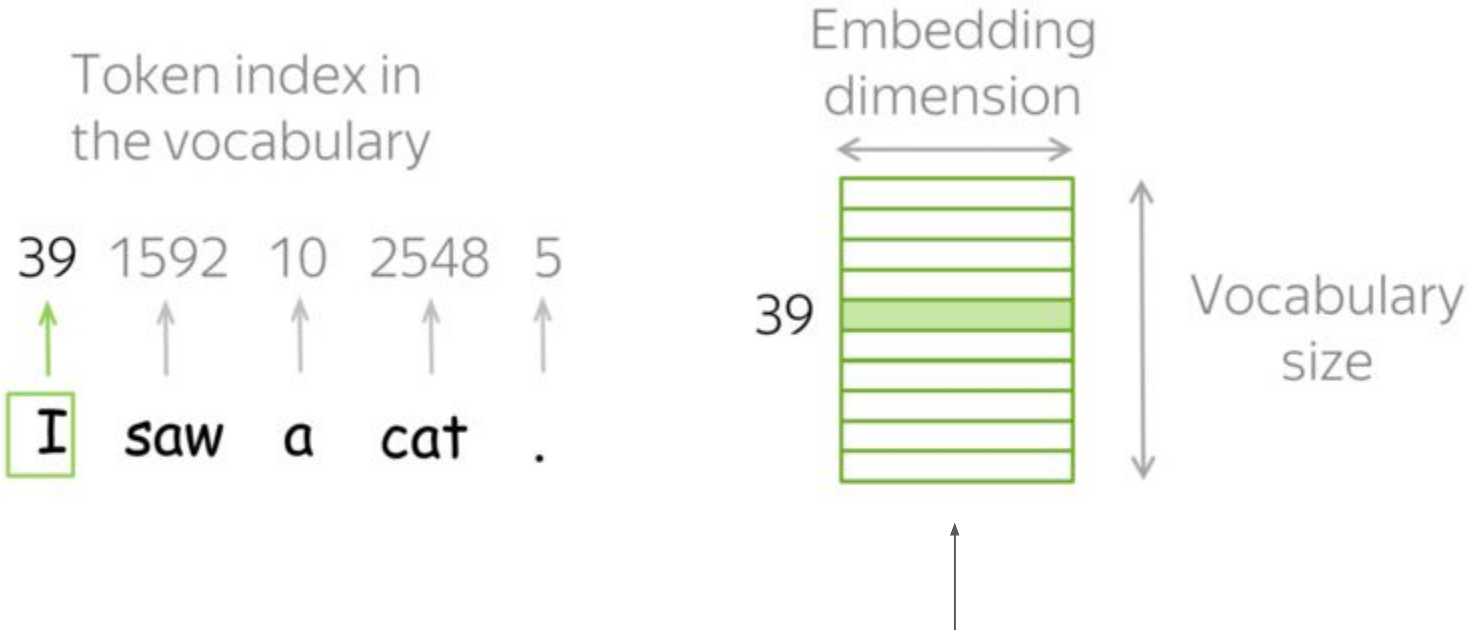
Any algorithm for solving a task

Word representation - vector
(input for your model/algorithm)

Sequence of tokens

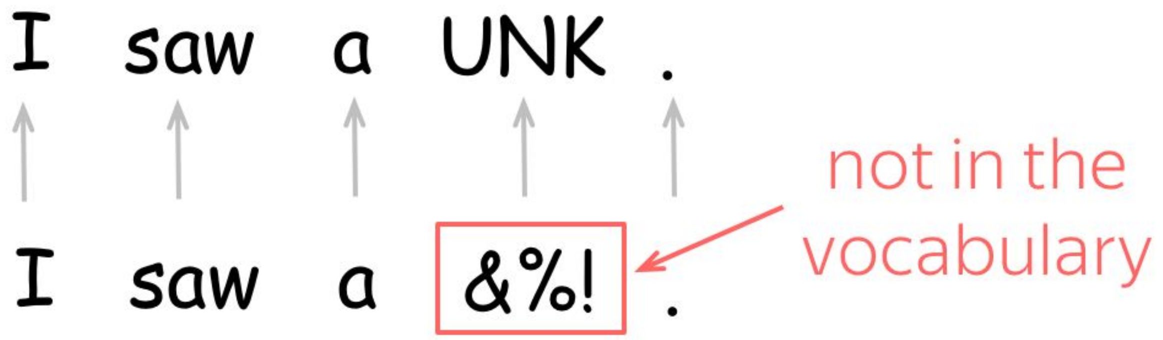
Text (your input)

Как работаем с эмбедингами?



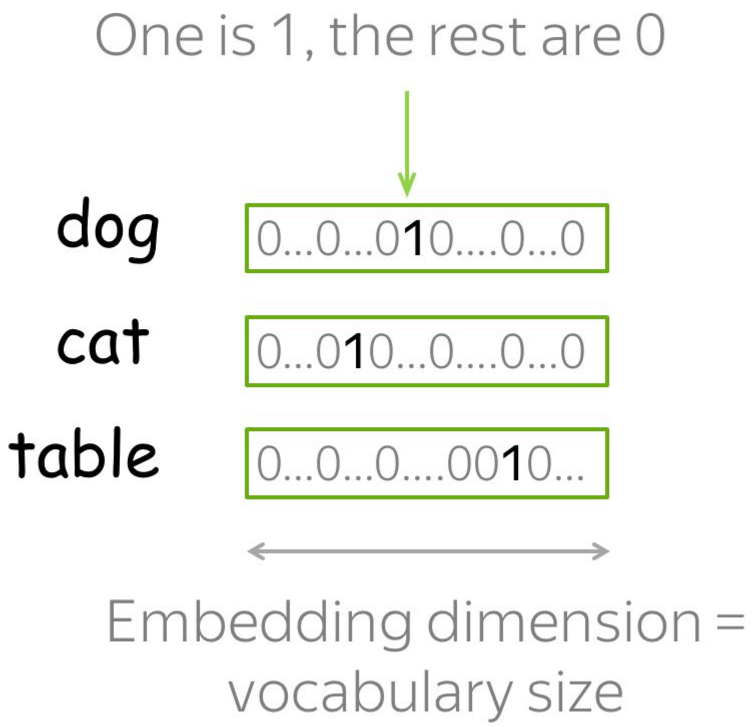
“Look-up” table

Что делать с незнакомыми словами?



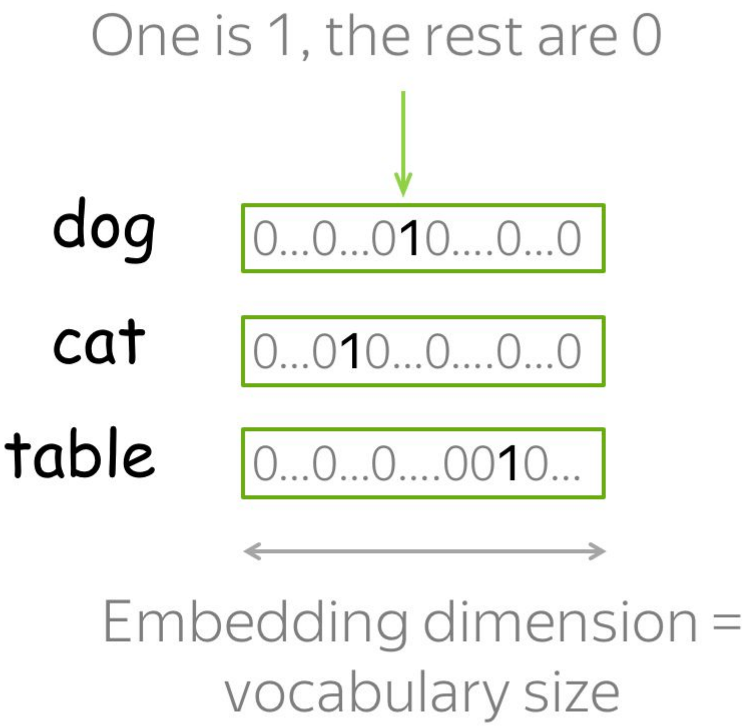
Embeddings

One-hot encoding



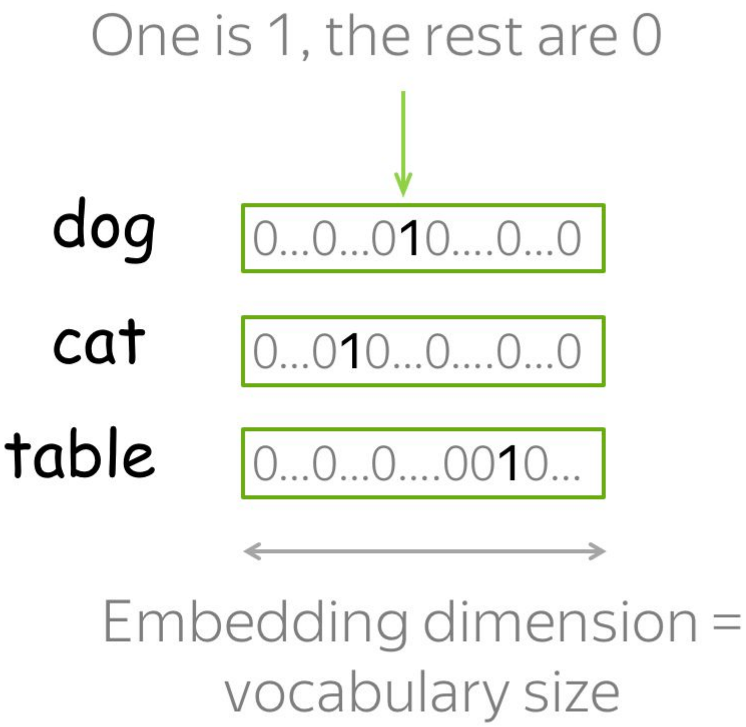
В чём проблема?

One-hot encoding



- размер эмбединга очень большой
- не отражают смысл слов

One-hot encoding



- размер эмбединга очень большой
- не отражают смысл слов

Смысл

Do you know what the word tezgüino means ?

(We hope you do not)

Смысл

Now look how this word is used in different contexts:

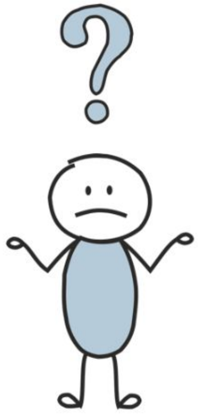
A bottle of tezgüino is on the table.

Everyone likes tezgüino.

Tezgüino makes you drunk.

We make tezgüino out of corn.

Can you understand what tezgüino means ?



Смысл

Now look how this word is used in different contexts:

A bottle of tezgüino is on the table.

Everyone likes tezgüino.

Tezgüino makes you drunk.

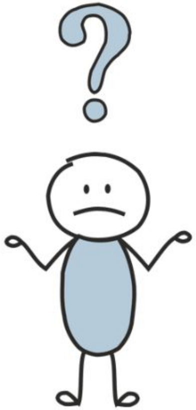
We make tezgüino out of corn.

Tezgüino is a kind of alcoholic beverage made from corn.

With context, you can understand the meaning!

Смысл

How did you do this?



Смысл

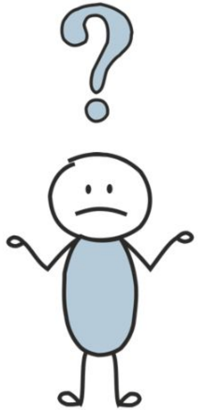
(1) A bottle of _______ is on the table.

(2) Everyone likes _______ .

(3) _______ makes you drunk.

(4) We make _______ out of corn.

What other words fit into these contexts ?



Смысл

(1) A bottle of _______ is on the table.

(2) Everyone likes _______ .

(3) _______ makes you drunk.

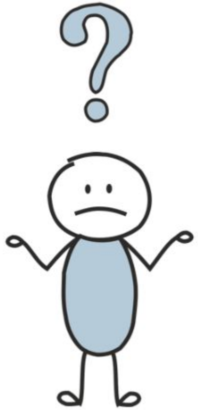
(4) We make _______ out of corn.

What other words fit into these contexts ?

	(1)	(2)	(3)	(4)	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
wine	1	1	1	0	

← contexts

← rows show contextual properties: 1 if a word can appear in the context, 0 if not



Смысл

(1) A bottle of _______ is on the table.

(2) Everyone likes _______ .

(3) _______ makes you drunk.

(4) We make _______ out of corn.

	(1)	(2)	(3)	(4)	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
wine	1	1	1	0	

rows are similar

Смысл

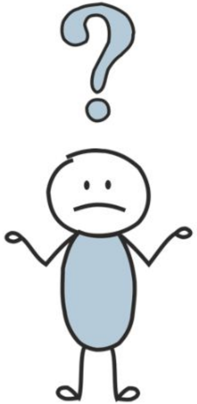
(1) A bottle of _______ is on the table.

(2) Everyone likes _______ .

(3) _______ makes you drunk.

(4) We make _______ out of corn.

	(1)	(2)	(3)	(4)	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
wine	1	1	1	0	



rows are similar → meanings of the words are similar

Is this true?

Смысл

(1) A bottle of _______ is on the table.

(2) Everyone likes _______ .

(3) _______ makes you drunk.

(4) We make _______ out of corn.

	(1)	(2)	(3)	(4)	...
tezgüino	1	1	1	1	
loud	0	0	0	0	
motor oil	1	0	0	1	
tortillas	0	1	0	1	
wine	1	1	1	0	

This is the **distributional hypothesis**

rows are similar → meanings of the words are similar

Дистрибутивная гипотеза

> Words which frequently appear in **similar contexts** have **similar meaning**.

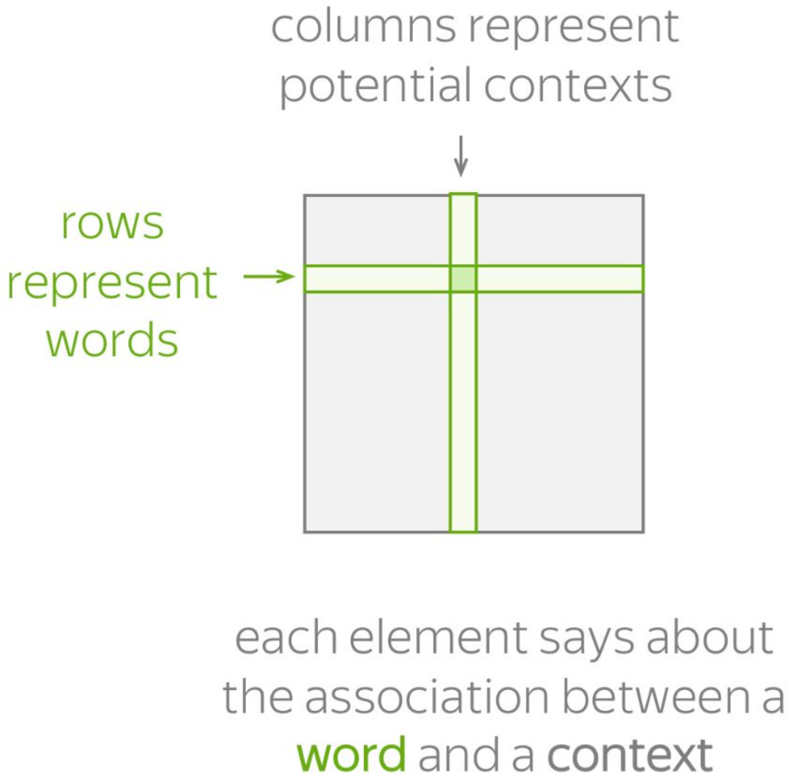
(Harris 1954, Firth 1957)

<u>Main idea:</u>

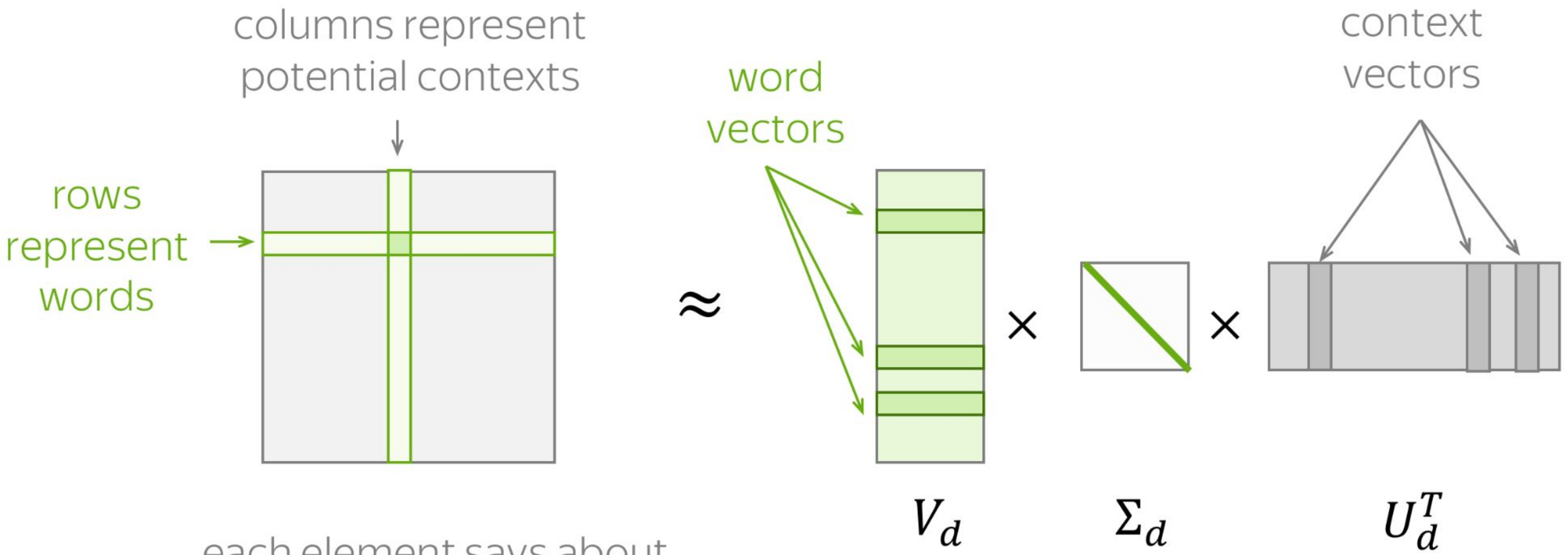
> We have to put information about contexts into word vectors.

What comes next: different ways to do this

Count based


columns represent
potential contexts
rows
represent
words
each element says about
the association between a
word and a context

Count based



each element says about the association between a **word** and a **context**

Reduce dimensionality:
Truncated Singular Value Decomposition (SVD)

TF-IDF

Context:

- document d (from a collection D)

Matrix element:

- tf-idf(w, d, D) = tf(w, d) · idf(w, D)

$$N(w, d) \qquad \log \frac{|D|}{|\{d \in D\colon w \in d\}|}$$

term frequency inverse document frequency

Word2Vec

Word2Vec

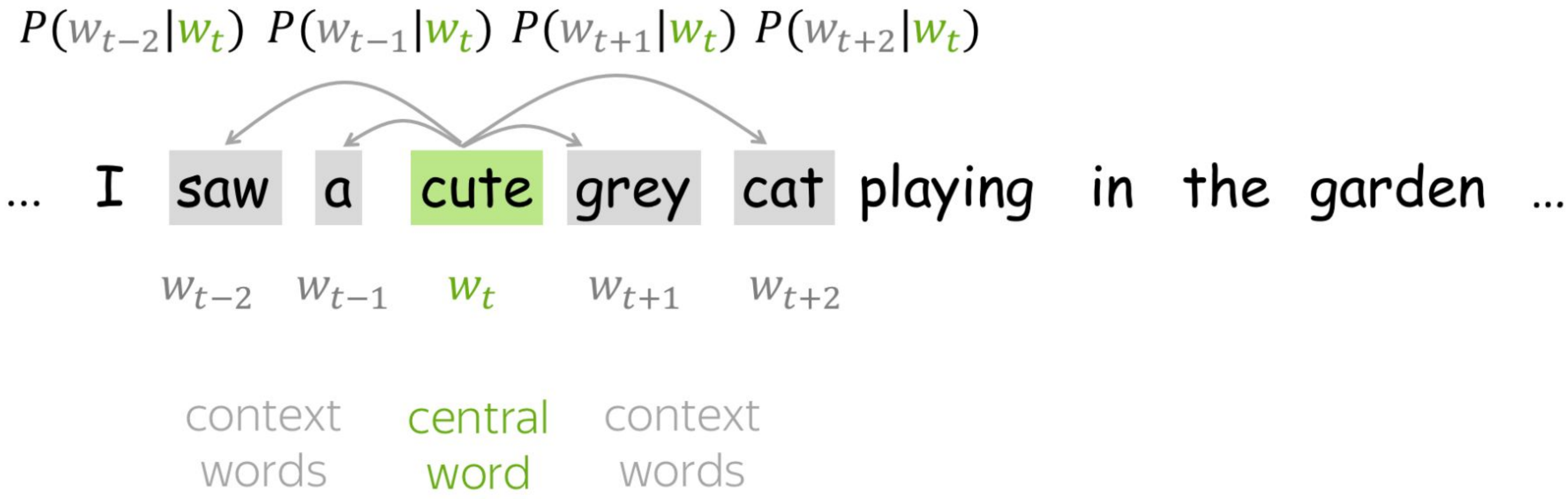
$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$

… I saw a cute grey cat playing in the garden …

w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}

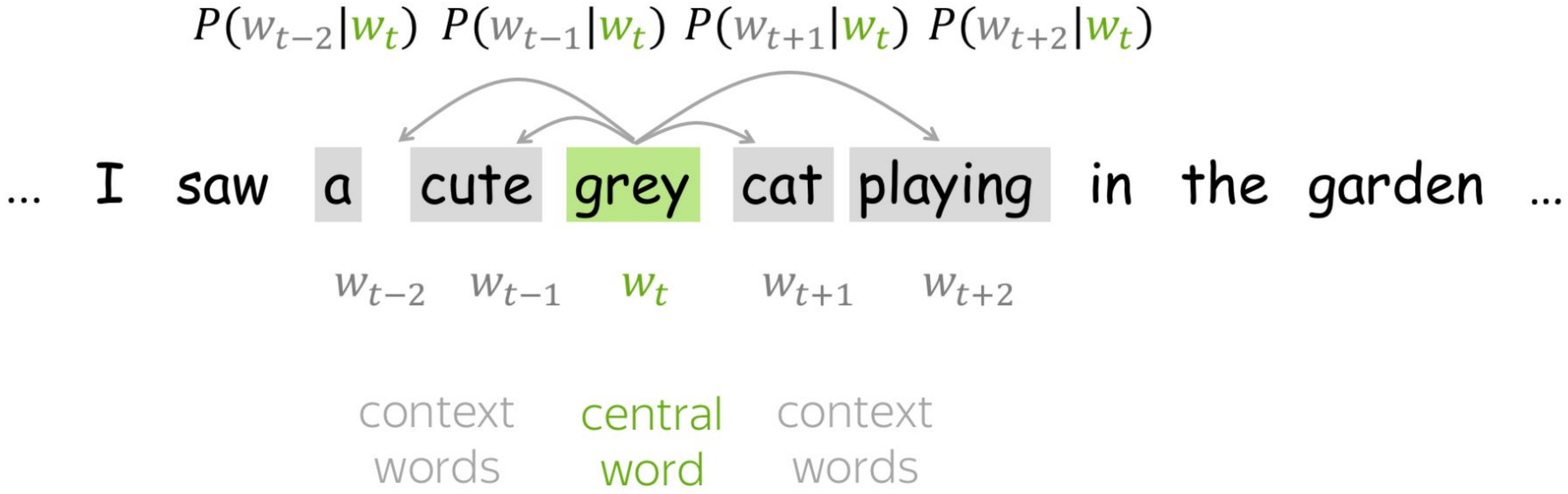
context words | central word | context words

Word2Vec



$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$
... I saw a cute grey cat playing in the garden ...
w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}
context words
central word
context words

Word2Vec



$P(w_{t-2}|w_t)$ $P(w_{t-1}|w_t)$ $P(w_{t+1}|w_t)$ $P(w_{t+2}|w_t)$
... I saw a cute grey cat playing in the garden ...
w_{t-2} w_{t-1} w_t w_{t+1} w_{t+2}
context words
central word
context words

Objective

Word2Vec tries to find the parameters that maximize the data likelihood:

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

We want our model to think that the training data is "likely"

To do this, it uses negative (log-)likelihood as its loss function:

$$\text{Loss} = J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m, \\ j \ne 0}} \log P(w_{t+j} | w_t, \theta)$$

agrees with our plan above ⟼ go over text

with a sliding window

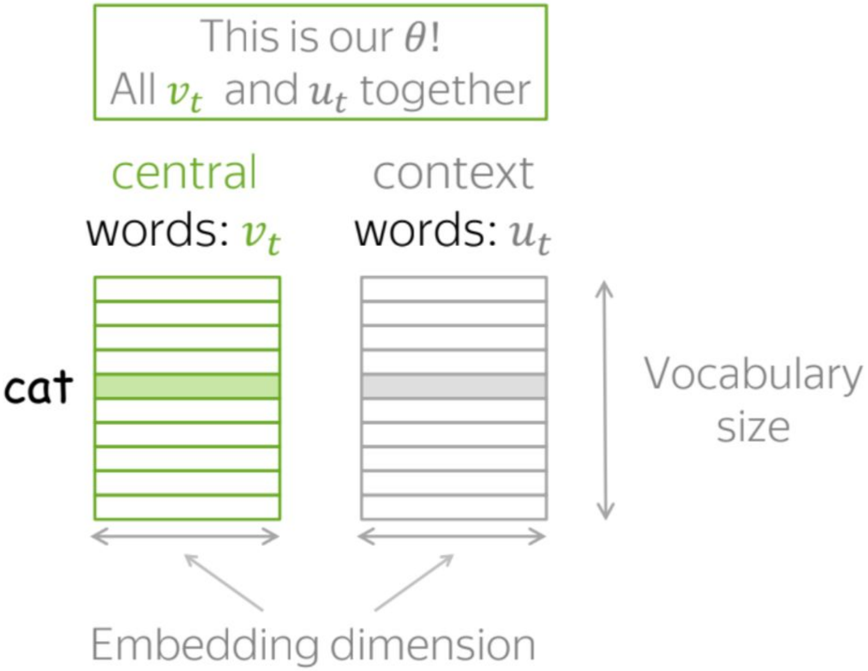
compute probability of the context word given the central

How to compute $P(w_{t+j}|w_t, \theta)$?

For each word w, we will have two vectors:

- v_w when it is a central word
- u_w when it is a context word

Once the vectors are trained, usually we throw away context vectors and use only word vectors.



How to compute $P(w_{t+j}|w_t,\theta)$?

For the central word c and context word o (o - outside):

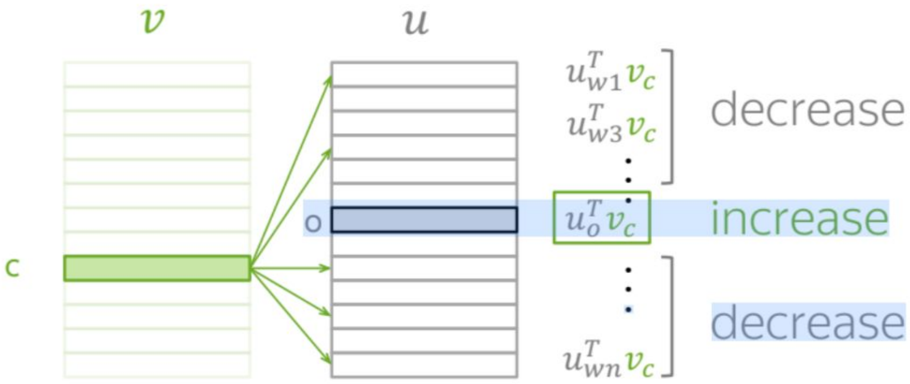
Dot product: measures similarity of o and c
Larger dot product = larger probability

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

Normalize over entire vocabulary
to get probability distribution

Let us recall our plan:

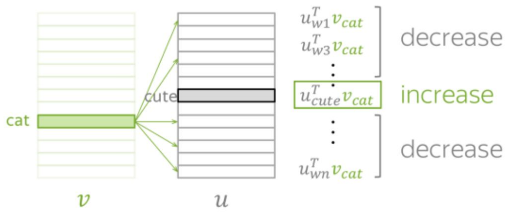
- ...
- adjust the vectors to increase these probabilities.



Negative sampling

Dot product of v_{cat}:

- with u_{cute} - increase,
- with <u>all other</u> u - decrease



Parameters to be updated:

- v_{cat}
- u_w for all w in the vocabulary

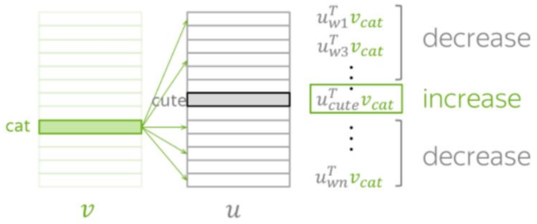
|V| + 1 vectors

Many parameters at each step – slow training

Negative sampling

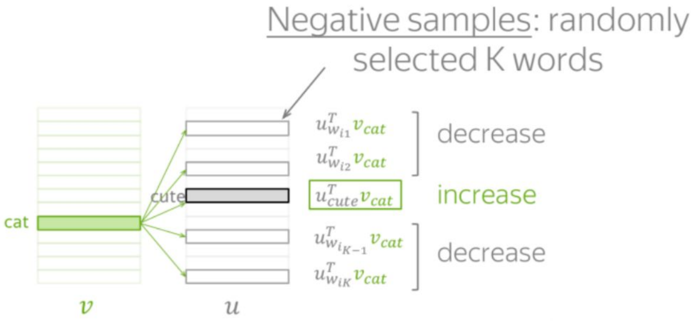
Dot product of v_{cat}:

- with u_{cute} - increase,
- with <u>all other</u> u - decrease



Dot product of v_{cat}:

- with u_{cute} - increase,
- with <u>a subset of other</u> u - decrease



Parameters to be updated: bad

- v_{cat}
- u_w for all w in the vocabulary

|V| + 1 vectors

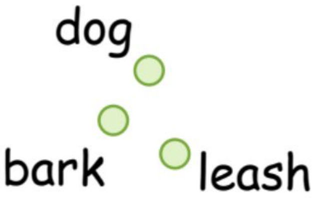
Parameters to be updated: good

- v_{cat}
- u_{cute} and u_w for w in K negative examples

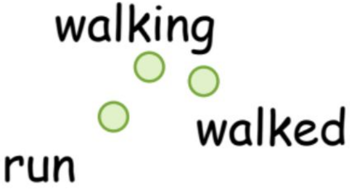
K + 2 vectors

Window size

- Larger windows – more topical similarities

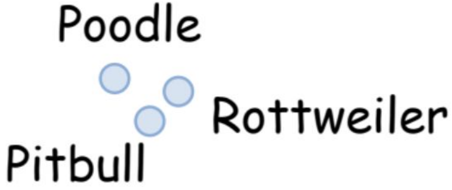


(grouped together)



(grouped together)

- Smaller windows – more functional and syntactic similarities



(grouped together)

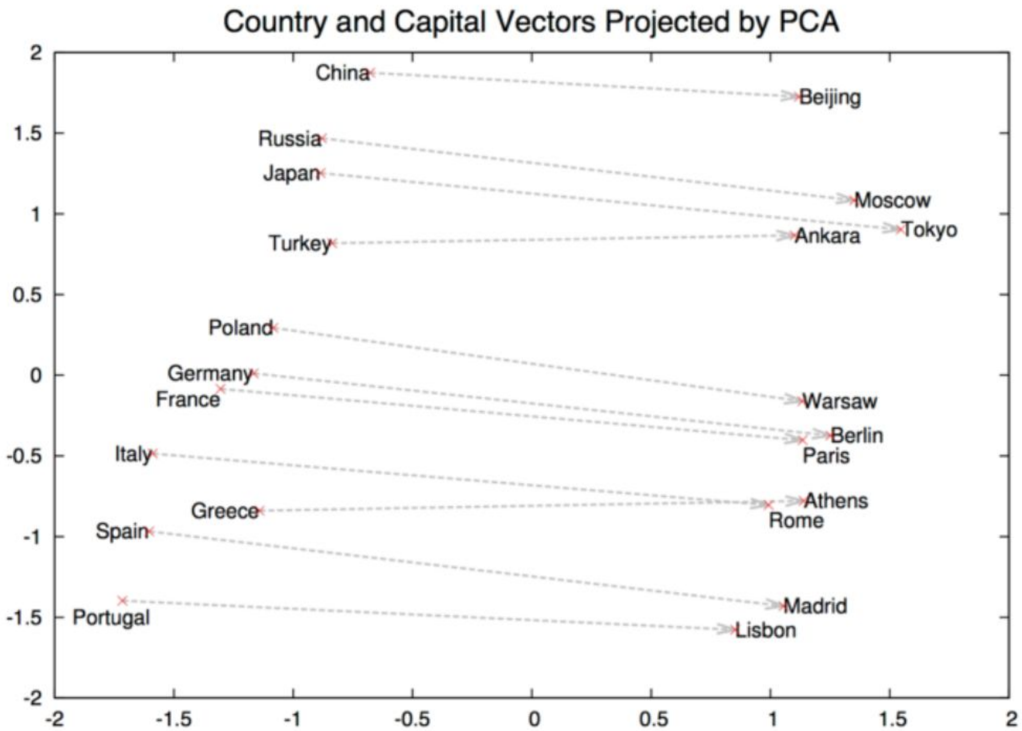
(grouped together)

Linear structure

semantic: $v(\mathbf{king}) - v(\mathbf{man}) + v(\mathbf{woman}) \approx v(\mathbf{queen})$

syntactic: $v(\mathbf{kings}) - v(\mathbf{king}) + v(\mathbf{queen}) \approx v(\mathbf{queens})$

Linear structure



Word2Vec