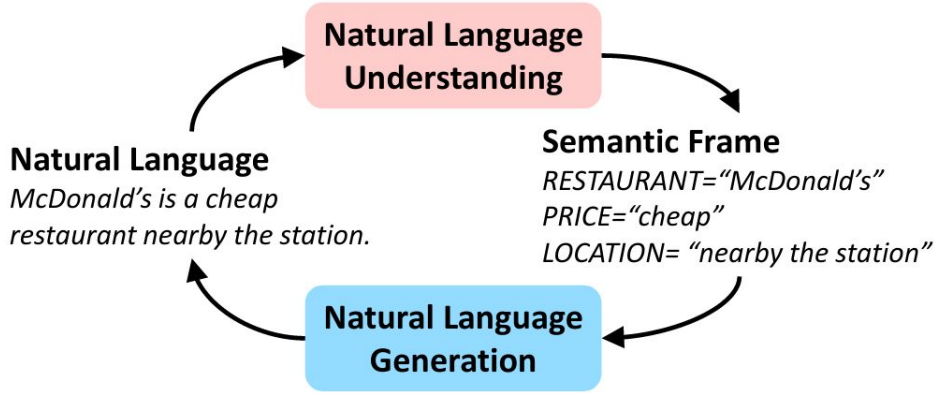
Natural Language Processing

Что такое Natural Language Processing?

```
nlp = linguistic + machine_learning
```

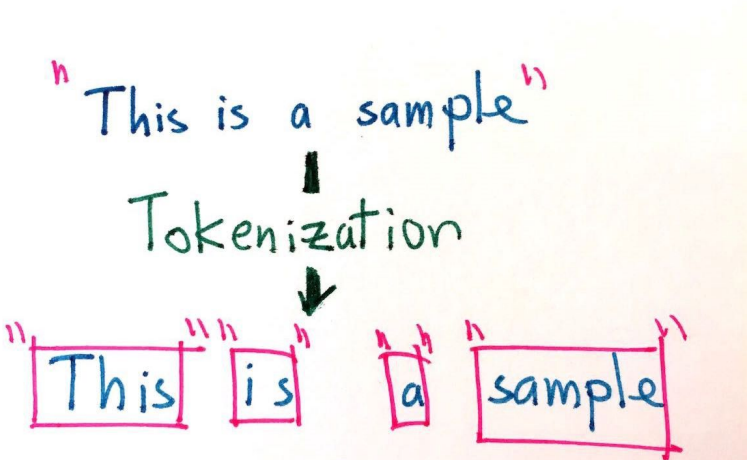
Структура

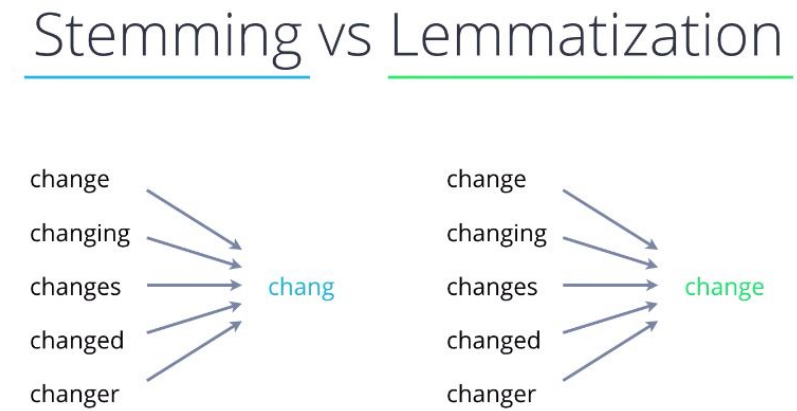


https://arxiv.org/pdf/1905.06196.pdf

Предобработка текста

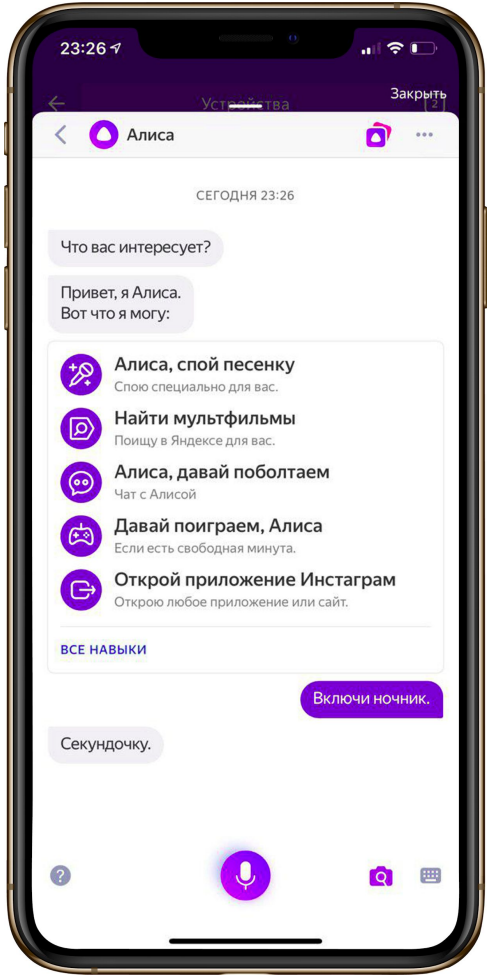
- Токенизация
- Нормализация слова
  - Стемминг
  - Лемматизация
- Удаление слов
  - Стоп-слова
  - Неинформативные слова, шаблоны





Примеры задач NLP

- Классификация текста: жанр, автор текста и т.п.
- Исправление опечаток
- Ранжирования поисковой выдачи
- Генерация текста, картинки по тексту
- Машинный перевод
- Диалоговые системы
- Суммаризация текста

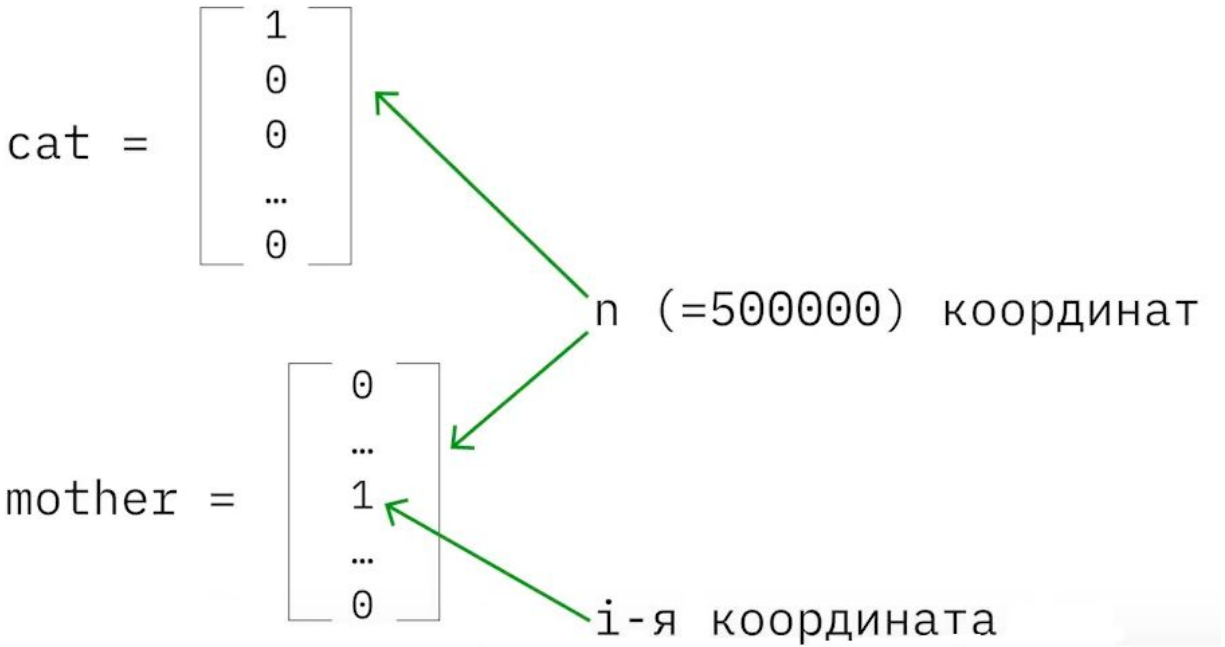


Выделение признаков

Какое компьютерное представление слов может быть?

Простой способ: one-hot encoding

One-hot вектор размерности длины словаря (например, 500.000)



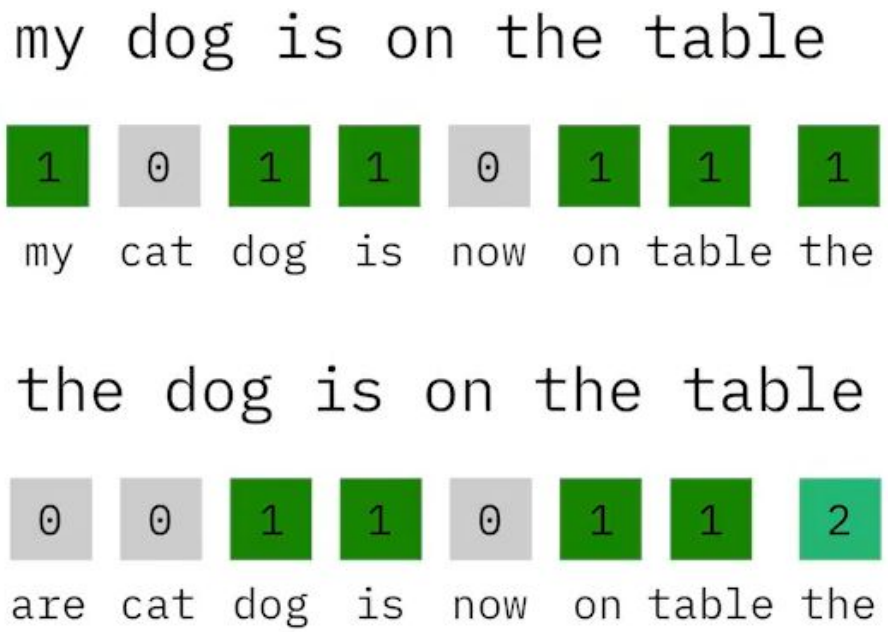
One-hot encoding

Проблемы:

- вектора не отображают значения слов
- не установить меру близости между словами
- большой размер векторов при малом количестве содержимой информации

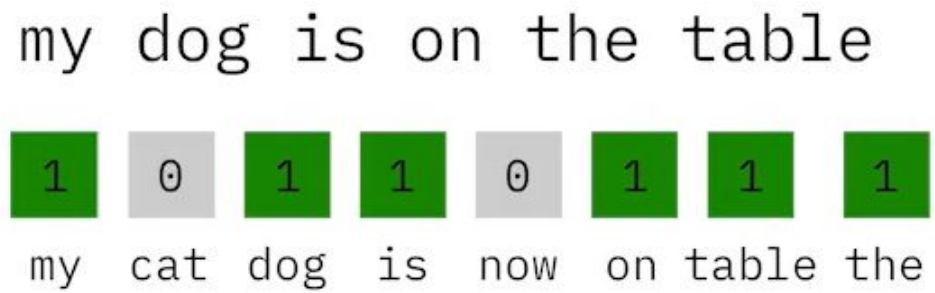
Bag-of-Words

Сумма one-hot векторов слов



Bag-of-Words

Сумма one-hot векторов слов



+ Проблема:
- нет информации о порядке слов

TF-IDF

n_{dw} - число вхождений слова w в документ d;

N_{w} - число документов, содержащих w;

N - число документов;

$p(\mathrm{w}, \mathrm{d}) = N_{\mathrm{w}}/N$ - вероятность встретить слово w в любом документе d

TF-IDF

$P(\mathrm{w}, \mathrm{d}, n_{\mathrm{dw}}) = (N_{\mathrm{w}}/N)^{n_{\mathrm{dw}}}$ - вероятность встретить n_{dw} раз слово w в документе d

$$-\log P(\mathrm{w}, \mathrm{d}, n_{\mathrm{dw}}) = n_{\mathrm{dw}} \cdot \log\left(N/N_{\mathrm{w}}\right) = TF(\mathrm{w}, \mathrm{d}) \cdot IDF(\mathrm{w})$$

$TF(\mathrm{w}, \mathrm{d}) = n_{\mathrm{dw}}$ - term frequency;
$IDF(\mathrm{w}) = \log\left(N/N_{\mathrm{w}}\right)$ - inverted document frequency;

TF-IDF

$$\text{tf}(''\text{this}'', d_1) = \frac{1}{5} = 0.2$$

$$\text{tf}(''\text{this}'', d_2) = \frac{1}{7} \approx 0.14$$

$$\text{idf}(''\text{this}'', D) = \log\left(\frac{2}{2}\right) = 0$$

$$\text{tfidf}(''\text{this}'', d_1, D) = 0.2 \times 0 = 0$$

$$\text{tfidf}(''\text{this}'', d_2, D) = 0.14 \times 0 = 0$$

Word ‘this’ is not very informative

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

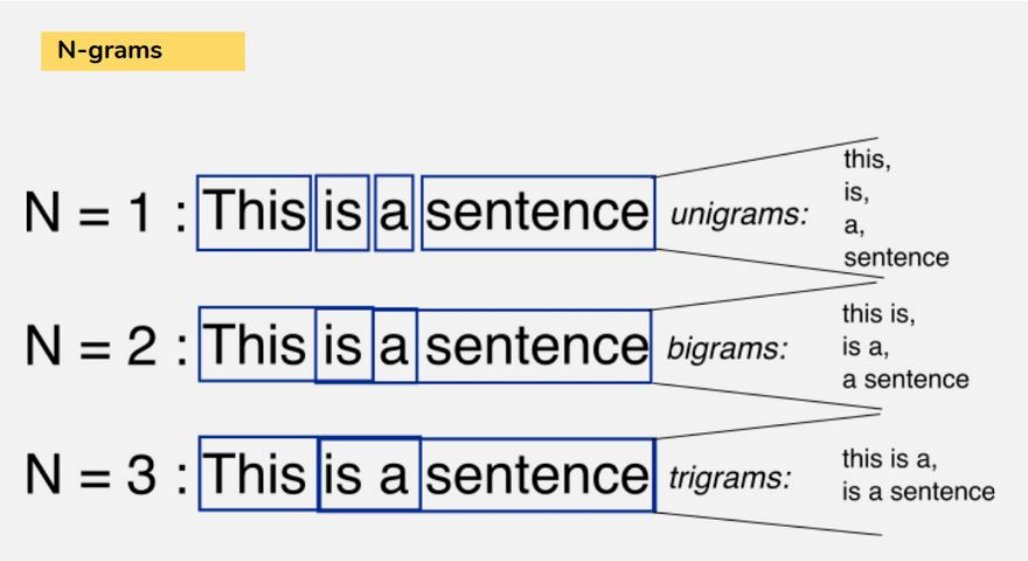
Term	Term Count
this	1
is	1
another	2
example	3

https://github.com/girafe-ai/ml-mipt/blob/master/week0_08_extra_texts/ml-mipt_Fall_2019_Lecture08_WordVectors.pdf

Коллокация

N-грамма - последовательность из N слов, идущих подряд

Коллокация - сочетание слов, не обязательно идущих подряд



Pointwise mutual information

Скользящие окно фиксированной длины:

n_{uv} - встречаемость слова u и v вместе

$$PMI = \log \frac{p(u,v)}{p(u)p(v)} = \log \frac{n_{uv} n}{n_u n_v}$$

$$pPMI = \max(0, PMI)$$

Pointwise mutual information

Word 1	Word 2	CountWord 1	CountWord 2	Count of co-occurrences	PMI
puerto	rico	1938	1311	1159	10.0349081703
hong	kong	2438	2694	2205	9.72831972408
los	angeles	3501	2808	2791	9.56067615065
carbon	dioxide	4256	1353	1032	9.09852946116
prize	laureate	5131	1676	1210	8.85870710982
san	francisco	5237	2477	1779	8.83305176711
nobel	prize	4098	5131	2498	8.68948811416

More clever way: context embeddings

Let’s consider words that can fit in the gaps:

1. Marie rode a _____
2. _____ wheel was punctured
3. The _____ has a beautiful white frame

	1	2	3
Bicycle	+	+	+
Bike	+	+	+
Car	+	+	-
Horse	+	-	-

Context embeddings

Let’s take into account words meanings in some way:

$v(word_i)[\,j\,]$ = count(co-occurrences $word_i$ with $word_j$ in dataset)

$v(word_1)$ = [12, 1, 0, 10, 5,...]

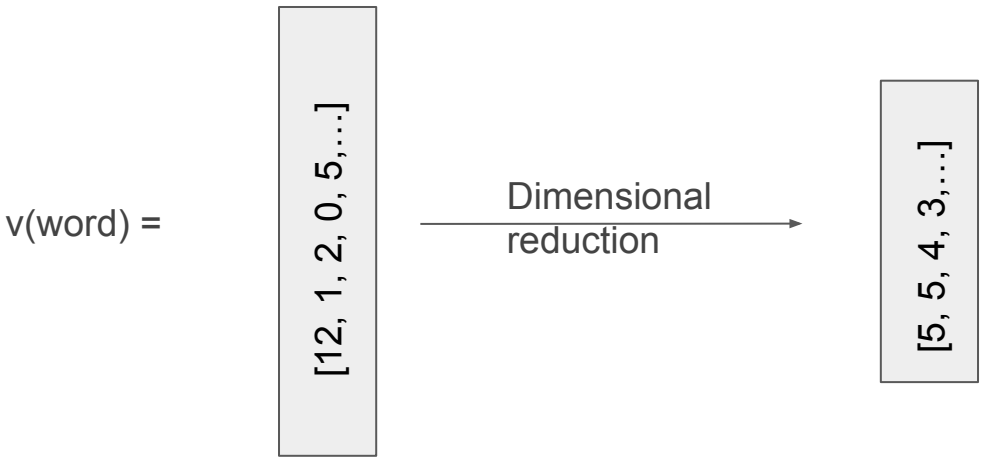
horse ride wheel roof hair breed

$v(word_2)$ = [20, 10, 0, 0, 1,...]

car ride wheel roof hair breed

Context embeddings

v(word$_i$)[j] = count(co-occurrences word$_i$ with word$_j$ in dataset)

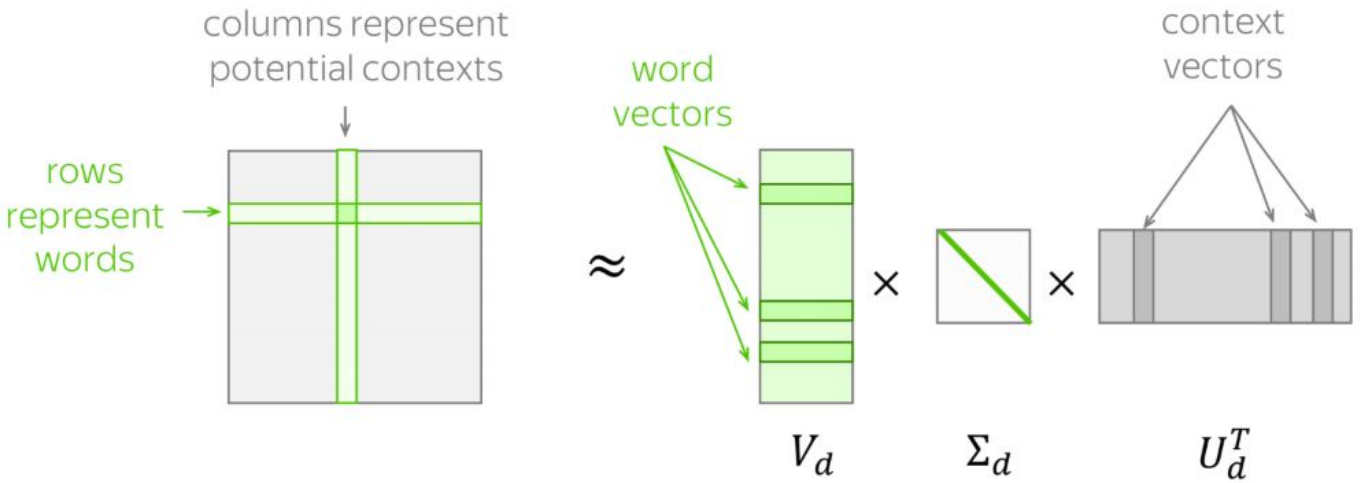


Context embeddings

Проблемы:

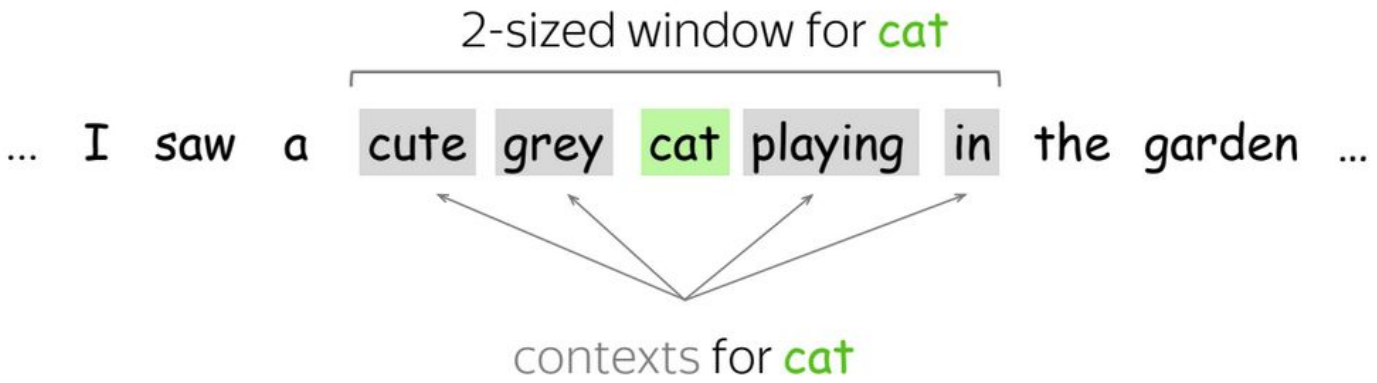
- редкие слова
- требуются много вычислительных ресурсов
- при изменении датасета требуются новые вычисления

Singular value decomposition



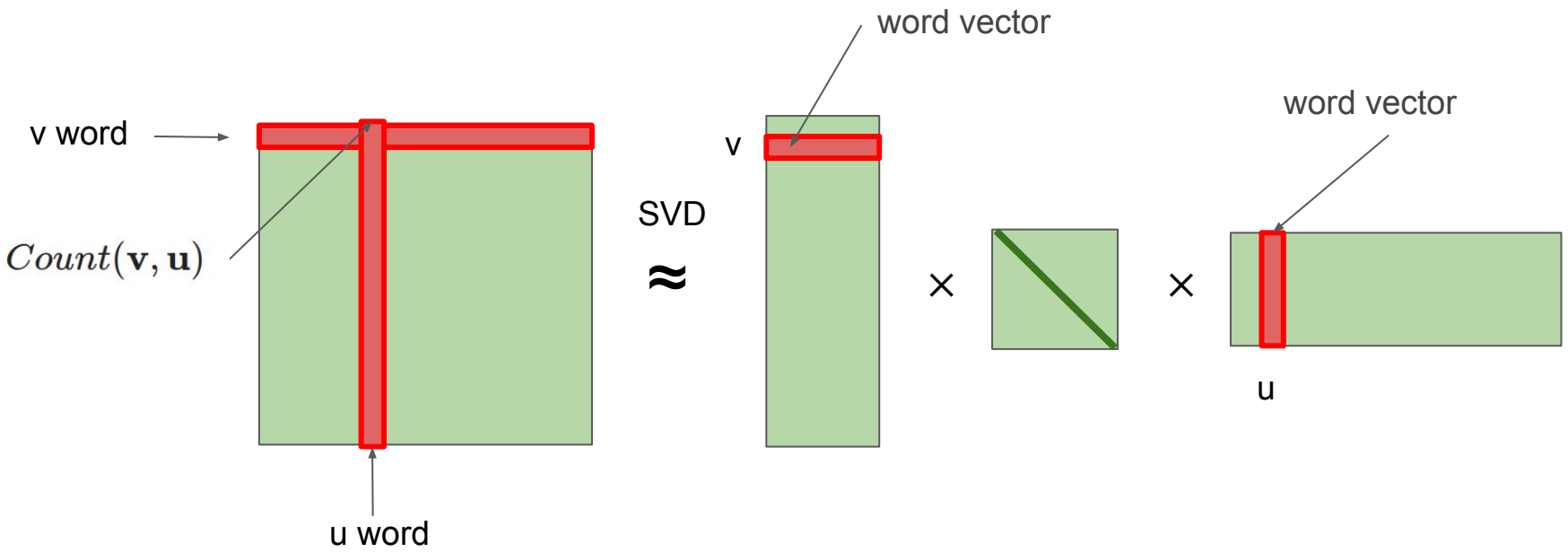
https://drive.google.com/file/d/1y2GKIKBzie7l8iycBO6gTKGiTTfJc4Dr/view

Co-Occurrence Counts

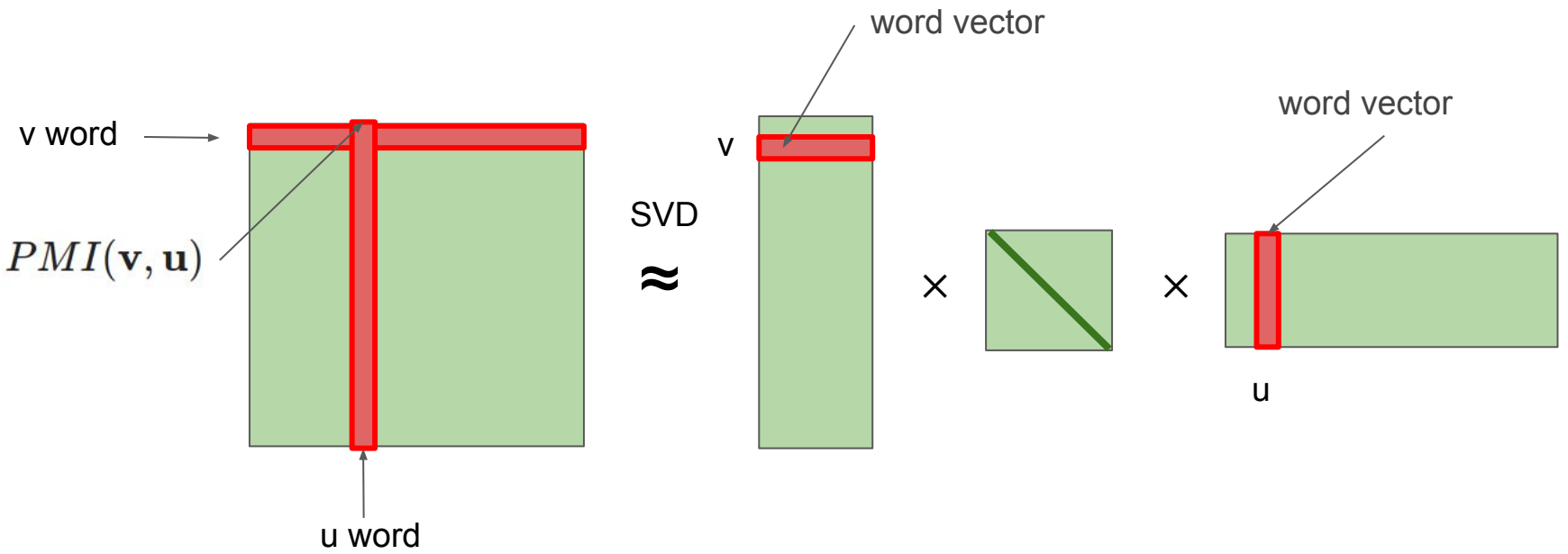


https://drive.google.com/file/d/1y2GKIKBzie7l8iycBO6gTKGiTTfJc4Dr/view

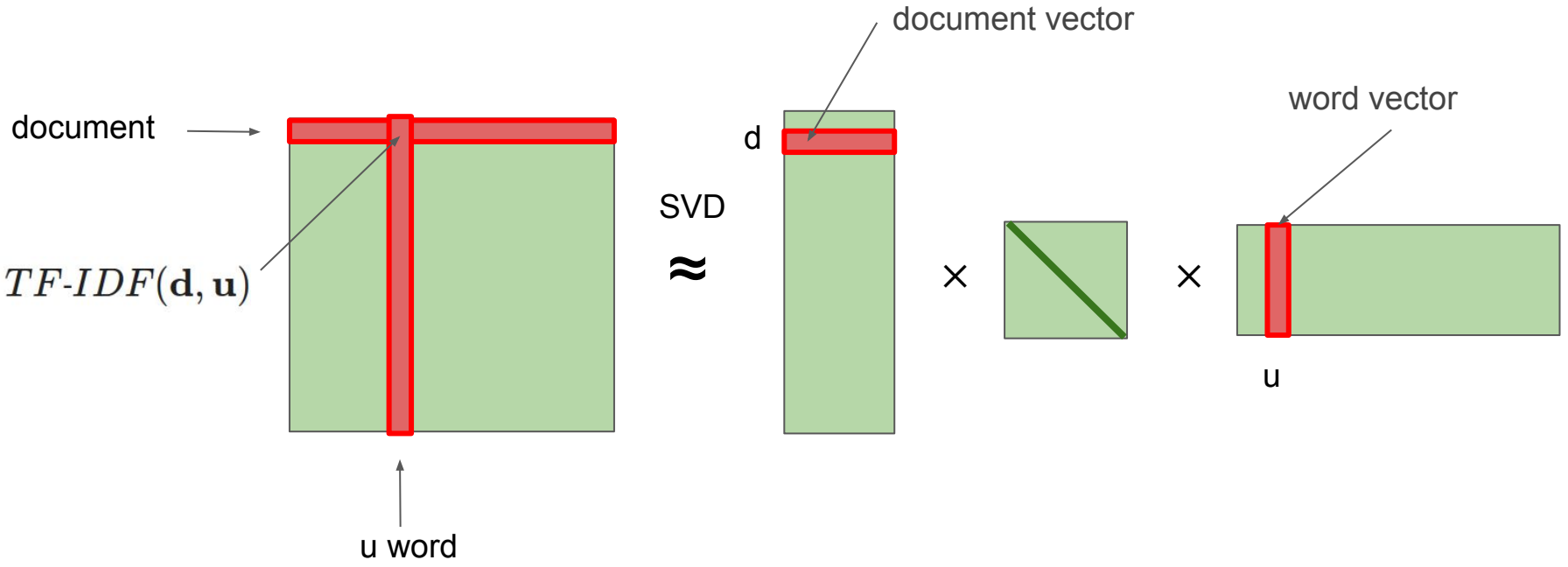
Co-Occurrence Counts



Pointwise mutual information



Latent semantic analysis



Классификация текстов

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
  - Спам-фильтр

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
  - Спам-фильтр
  - Токсичные комментарии

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
  - Спам-фильтр
  - Токсичные комментарии
  - Фейк-ньюс

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
  - Спам-фильтр
  - Токсичные комментарии
  - Фейк-ньюс
- Не совсем классификация, а скорее регрессия
  - Оценка тональности текста

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
  - Спам-фильтр
  - Токсичные комментарии
  - Фейк-ньюс
- Не совсем классификация, а скорее регрессия
  - Оценка тональности текста
- Как часть другой задачи NLP
  - Фильтрация обучающей выборки

Зачем классифицировать текст?

- Просто чтобы классифицировать тексты
  - Спам-фильтр
  - Токсичные комментарии
  - Фейк-ньюс
- Не совсем классификация, а скорее регрессия
  - Оценка тональности текста
- Как часть другой задачи NLP
  - Фильтрация обучающей выборки
  - Выбор сценария в диалоговой системе

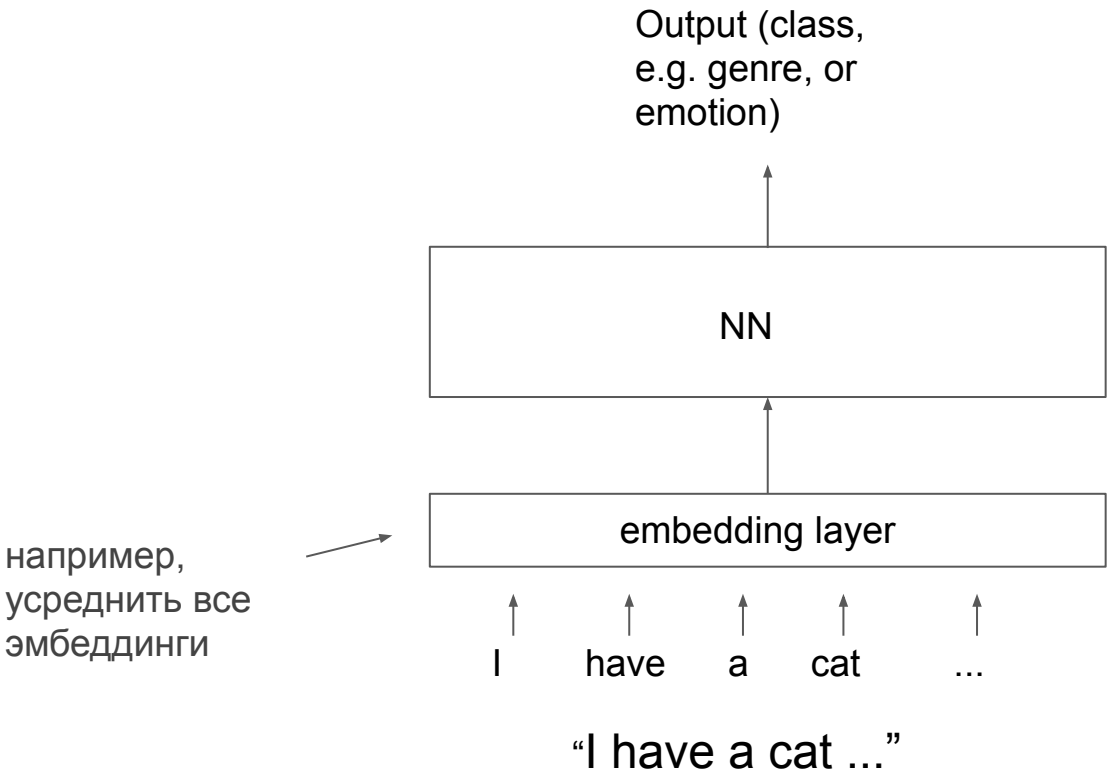
Методы классификации текстов

Классические методы

Признаки:

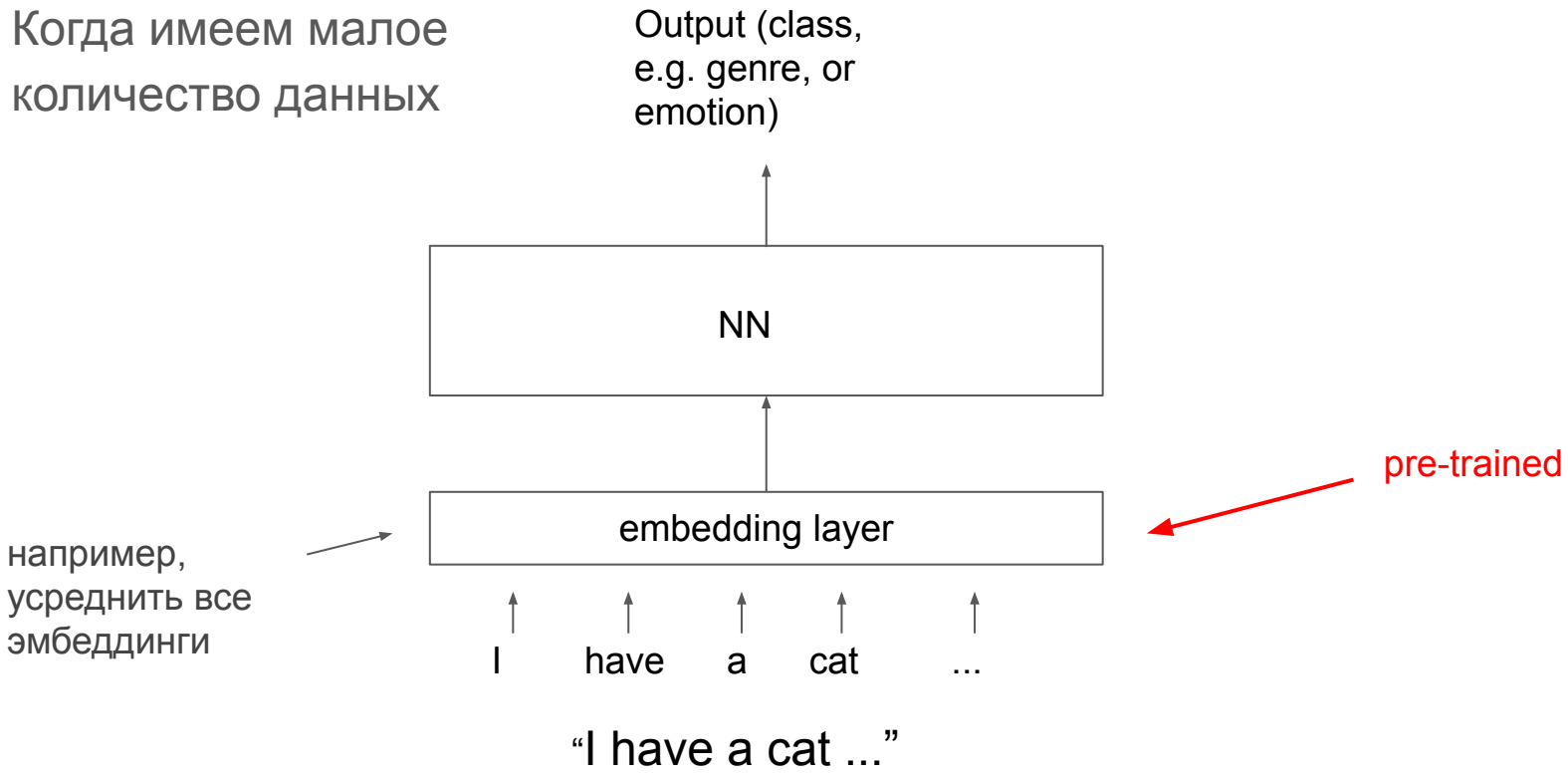
- Bag of Words
- Tf-Idf
- Embeddings

Задача классификации

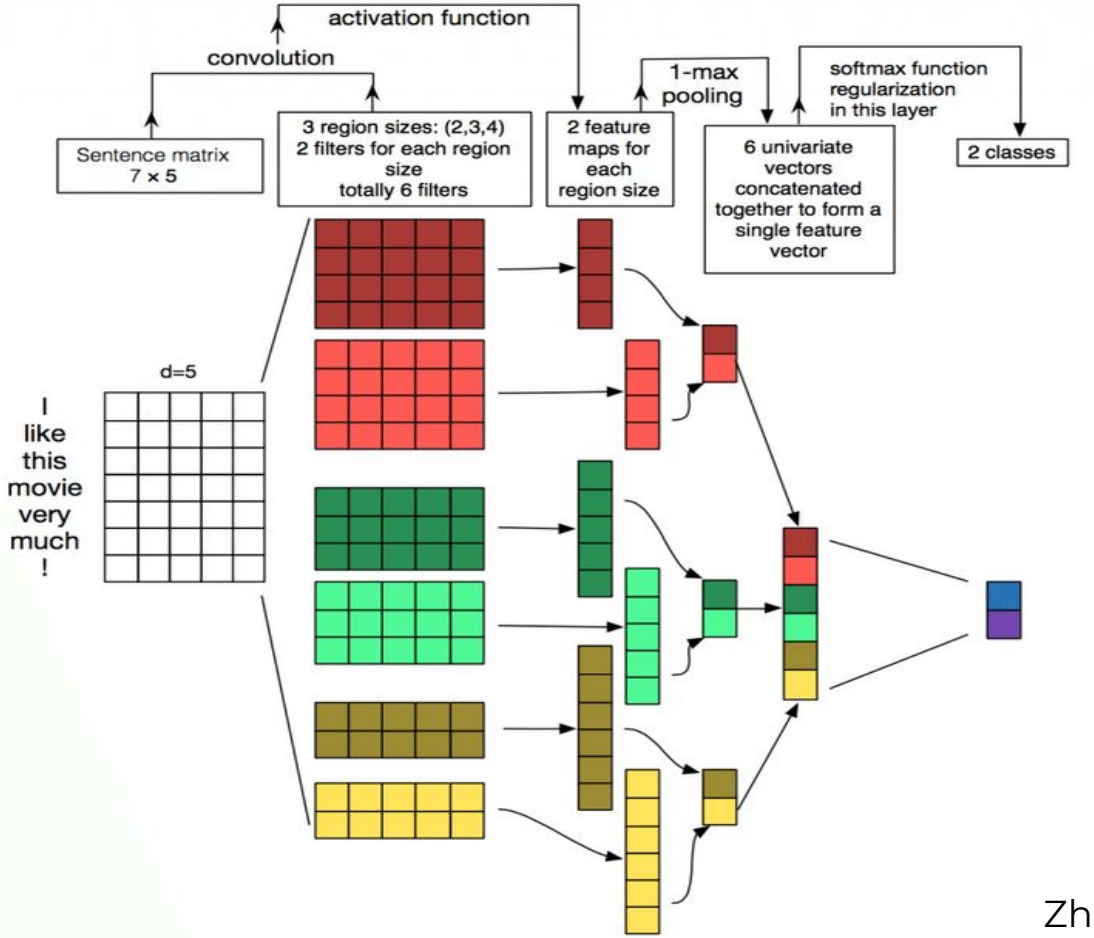


Задача классификации

Когда имеем малое количество данных



CNN



Zhang et al., 2015