

# ВЕКТОРИЗАЦИЯ ТЕКСТОВ

о бейзлайне для

Harry Potter and the Action Prediction  
Challenge from Natural Language

Есть текст фанфика по Поттеру.

В фанфике иногда происходят заклинания

**Задача:** по тексту, предшествующему заклинанию, угадать, какое заклинание сейчас будет.

Задача:

по тексту, предшествующему заклинанию, угадать, какое заклинание сейчас будет.

Наивный бейзлайн:

простой BoW вектор текста  
запихнуть в простой классификатор

Кусок статьи  
про датасет

### 3.1 Machine learning models

The input sentence  $w_{1:n}$  is encoded as a one-hot vector,  $\mathbf{v}$  (total occurrence weighting scheme).

**Multinomial Logistic Regression** Let  $\text{MLR}_\theta(\mathbf{v})$  be an abstraction of a multinomial logistic regression parametrized by  $\theta$ , the output for an input  $\mathbf{v}$  is computed as the  $\arg \max_{a \in A} P(y = a | \mathbf{v})$ , where  $P(y = a | \mathbf{v})$  is a *softmax* function, i.e.,  
$$P(y = a | \mathbf{v}) = \frac{e^{W_a \cdot \mathbf{v}}}{\sum_{a'} e^{W_{a'} \cdot \mathbf{v}}}.$$

простой BoW:  
count vectorizer

простой классификатор

как вообще давать тексты в модели

# как вообще давать тексты в модели

Подход, основанный на BoW  
превращении текстов в векторы

Подход, основанный на  
использовании готовых  
векторов — эмбеддингов

BoW

# Bag of words представление корпуса

$D$  — корпус,  $d_i$  — документ,  $t_{ij}$  — токен,

$|X|$  — количество элементов в  $X$ ,

$RN$  — пространство, элементы которого — упорядоченные наборы из  $N$  вещественных чисел (i.e. векторы в  $RN$ )

$D$  состоит из документов  $d_1, \dots, d_i, \dots, d_{|D|}$ ,

документ  $d_i$  состоит из токенов  $t_{i1}, \dots, t_{ij}, \dots, t_{i|d_i|}$

BoW (e.g. Tf-Idf и Count (частоты)) переводят элементы корпуса  $D$  в элементы пространства  $RN$ , где  $N$  — кол-во уникальных слов в корпусе



# ONE

Есть токенизированный, лемматизированный текст на русском

Есть словарь русского языка с пронумерованными словарными статьями

$\text{ONE}(\text{слово}) = V1 = (0, 0, 0, \dots, 0, 1, 0\dots)$ ,

- $|V1|$ =размер словаря,
- 1 на позиции, соответствующей номеру *слова* в словаре

# MNE

для текста.  $\text{MNE}(\text{текст}) = V2(\text{текст})$ ,  $|v2| = |v1|$ ,  $V2(\text{текст}) = (0, 1, 0, \dots, 0, 1, 0\dots)$ ,

для каждого слова текста поставлена величина на позиции токена в словаре

Один из способов получения  $\text{MNE}(\text{текст})$ : сложить ONE каждого его токена.

Вектор для индекса:

$(0, \dots, 0, 1, \dots, )$

и.д.б.т.  
и.е. 1

←————→  
# слов в словаре

Вектор для текста:

$(0, \dots, 0, 1, \dots, )$

м.быть  
и 1

А, напр. Tf-Idf-вес  
слова в тексте

←  $\#$  слов в словаре →

$$Tf_w \cdot idf = Tf_w \cdot \frac{1}{\frac{|dw|}{|D|}}$$

V — словарь,

w — слово

D — корпус,  
коллекция

текстов

|D| — число док-в в корпусе

|dw| — число док-в с словом w


$Tf_w$  — частота слова w в док-те.

# Векторные пространства

Векторное пространство неформально — куча штук (векторов)

- Векторы друг с другом можно складывать, получая векторы
- Векторы можно умножать на числа.
- Описанное работает привычно ( $v_1 + v_2 = v_2 + v_1$ ,  $1 * v_1 = v_1$  и т.д.)

аксиомы  
вект. пр-ва



Вектор в  $R^n$  можно записать столбцом из  $n$  чисел

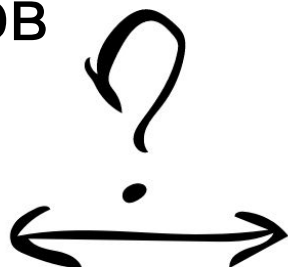
- Е.g. документ в Tf-Idf матрице — это такой вектор в  $R|D|$

Последовательность из  $m$  векторов в  $R^n$  можно записать в табличку-матрицу  $n \times m$  ( $n$  строк,  $m$  столбцов)

- Е.g. корпус в Tf-Idf представлении такая матрица в  $R|D|$

В  $R^n$  можно делать арифметику а ещё ввести расстояние между векторами

проблемы BoW векторов



The diagram shows a vector with a dot above it and a horizontal double-headed arrow below it. Below this is a matrix  $D$  with elements  $d_1, d_2, \dots$ . To the right of the matrix is a vertical double-headed arrow with the text "размер корпуса" (corpus size) next to it.

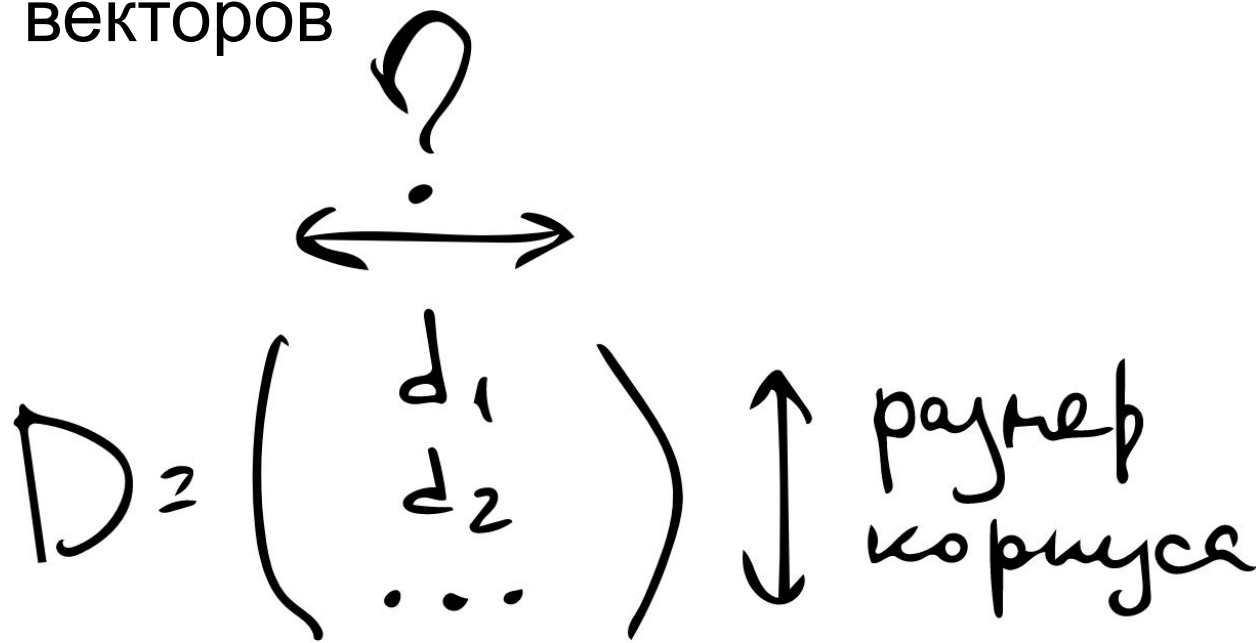
$$D = \begin{pmatrix} d_1 \\ d_2 \\ \dots \end{pmatrix}$$

размер корпуса

$d_i$  - го-т у корпуса  $D$

# проблемы BoW векторов

Они очень уж большие



A diagram illustrating the size of BoW vectors. It shows a matrix  $D$  with dimensions indicated by a double-headed arrow. The matrix is represented as  $D = \begin{pmatrix} d_1 \\ d_2 \\ \dots \end{pmatrix}$ . To the right of the matrix, a vertical double-headed arrow is labeled "размер корпуса" (corpus size). Above the matrix, a horizontal double-headed arrow is labeled with a question mark, indicating the large size of the vectors.

$$D = \begin{pmatrix} d_1 \\ d_2 \\ \dots \end{pmatrix} \quad \updownarrow \text{размер корпуса}$$

$d_i$  - doc-т из корпуса  $D$

# BoW векторы большие, что делать?

Хранить их разреженно

Брать в вектор только самые частотные слова

Снижать размерность



# Сингулярное разложение

Пусть есть матрица  $A$

Известно, что её можно представить

в виде  $A = L \cdot \Lambda \cdot R$  причём

↑            ↑            ↑  
матрица    матрица    матрица

- $\Lambda$  — диагональная матрица, элементы которой упорядочены по убыванию
- Зануление наименьших элементов  $\Lambda$  приводит к понижению размерности с минимальными потерями в дисперсии

на диагонали  
не нули

нули

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_n \end{pmatrix}$$

нули

# Методы снижения размерности

# LSI

Пусть корпус представлен BOW матрицей

Подход LSI заключается в том, чтобы

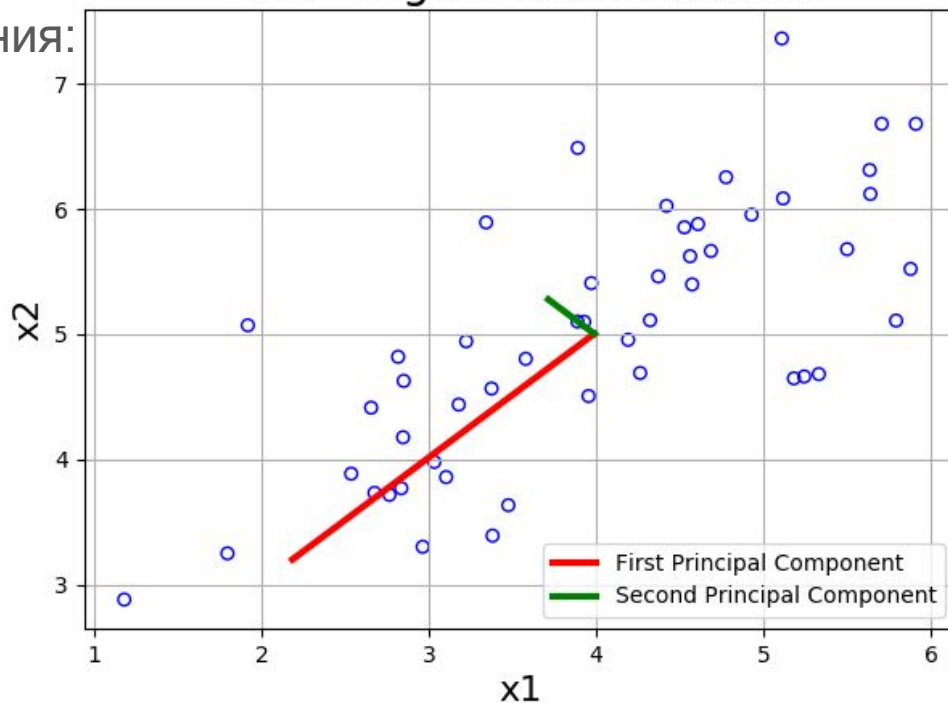
- сингулярно разложить BOW матрицу корпуса
- оставляя только  $k$  наибольших  $\lambda$  в срединной матрице разложения, делать снижение размерности
- получатся меньшие векторы документов

# PCA

Сначала центрируем векторы документов из корпуса, затем делаем LSI

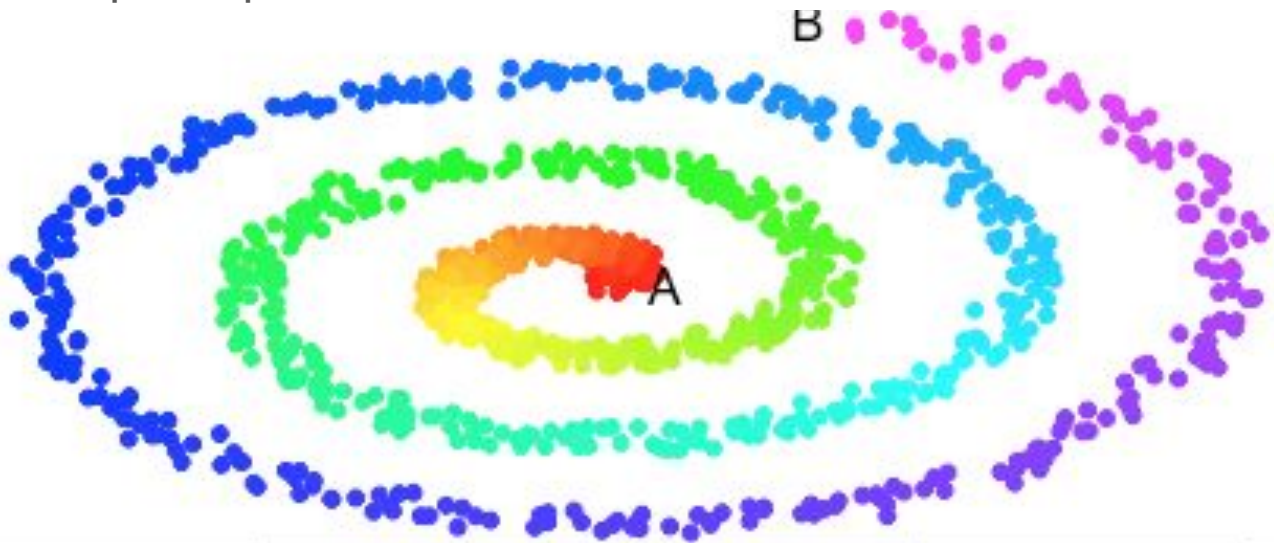
## PCA - Eigenvectors Shown

Смысл разложения:



# t-SNE

- Не строит линейное отображение
- Смотрит на отношения между точками для обработки нелинейностей
- Итеративно подбирает параметры



# UMAP

Как tSNE, но

- Моделирует распределение расстояний в оригинальном пространстве с помощью другого распределения
- По-другому предобрабатывает расстояния
- Моделирует оригинальное распределение расстояний не распределением Стюдента
- **Быстрее работает**

# pLSI

Вероятности из **ЗБЧ** и начальных приближений

Начальные приближения берутся из категориальных распределений, параметры которых подбираются итеративно.

$$P(D, W) = P(D) \sum_Z P(Z|D) P(W|Z)$$

$$P(D, W) = \sum_Z \underbrace{P(Z)}_{\text{blue}} \underbrace{P(D|Z)}_{\text{red}} \underbrace{P(W|Z)}_{\text{purple}}$$

$$A \approx \underbrace{U_t}_{\text{red}} \underbrace{S_t}_{\text{blue}} \underbrace{V_t^T}_{\text{purple}}$$

# LDA

Как pLSI, только начальные распределения генерируются из распределения Дирихле. Параметры распределения подбираются итеративно.





# LDA

Как pLSI, только начальные распределения генерируются из распределения Дирихле. Параметры распределения подбираются итеративно.



# BigARTM

Настраиваемое тематическое моделирование, SoTA



# Как ещё можно?

Эмбедингами.

# Эмбе́ддинги

Какой способ нахождения  
расстояния между эмбе́ддингами

лучше справится с похожестью  
текстов “КОТ, КОТ И КОТ” и “КОТ И КОТ”?

