



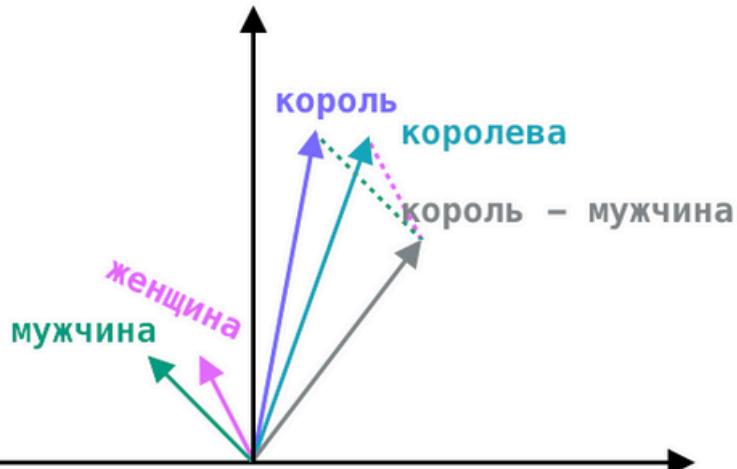
Факультет компьютерных
наук

Аналитика больших
данных

Москва
2025

Векторные представления слов и тематическое моделирование

Векторные представления и Topic Modeling



[Источник картинки](#)

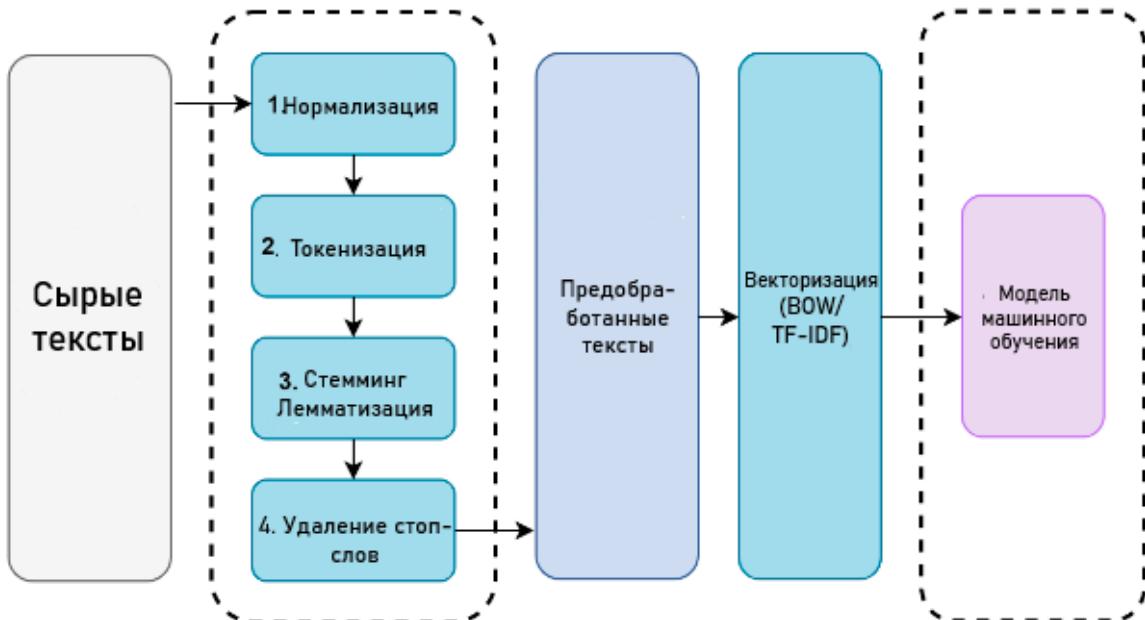
счетчик слов

представлениям с
геометрией смысла

[Linguistic Regularities in Continuous Space Word Representations](#)



1. Дистрибутивная гипотеза и распределительная семантика
2. Count-based vs prediction-based
3. GloVe: глобальные векторы слов
4. Нейросетевые эмбеддинги: Word2Vec
5. От слов к документам: идеи тематического моделирования



Ограничения BOW и TF-IDF:

- разреженность,
- отсутствие порядка слов и контекста;
- невозможность ловить синонимы / аналогии.



Просто добавь контекст?

- (1) Можно погладить ____.
(2) ____ лает.
(3) У ____ есть хвост.
(4) ____ живет дома.

	(1)	(2)	(3)	(4)	...
крякозябра	1	1	1	1	...
кошка	1	0	1	1	...
собака	1	1	1	1	...
корова	0	0	1	0	...
попугай	1	0	1	1	...
змея	0	0	1	0	...



строки
похожи

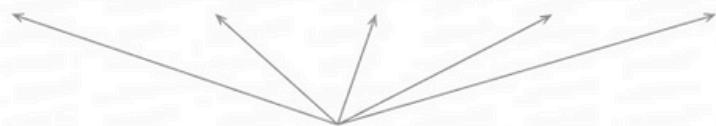
значения
слов похожи

You shall know a word by the company it keeps

J. R. (John Rupert) Firth

Окно размером 2 для слова "кошка"

... Я видел рыжую пушистую кошку, спящую на солнышке ...



контексты для слова "кошка"



Count-based vs prediction-based

Count-based (Статистические)	Prediction-based (Предсказательные)
Идея: Глобальная статистика	Идея: Локальный контекст
Строим матрицу "слово-контекст" (co-occurrence) и сжимаем её.	Не строим матрицу. Учим модель предсказывать слово или контекст.
Механизм:	Механизм:
1. Считаем частоты (Raw counts/PPMI). 2. Понижаем размерность (SVD, факторизация).	1. Скользящее окно. 2. Градиентный спуск обновляет веса. 3. Веса модели = эмбеддинги.
Примеры:	Примеры:
LSA/LSI, GloVe (смешанный тип).	Word2Vec (CBOW, Skip-gram), FastText.



Идея: сначала считаем, как слова совместно встречаются в корпусе (co-occurrence), получаем большую матрицу “слово-контекст”, а потом сжимаем/нормируем её, чтобы получить вектора.

Что считается:

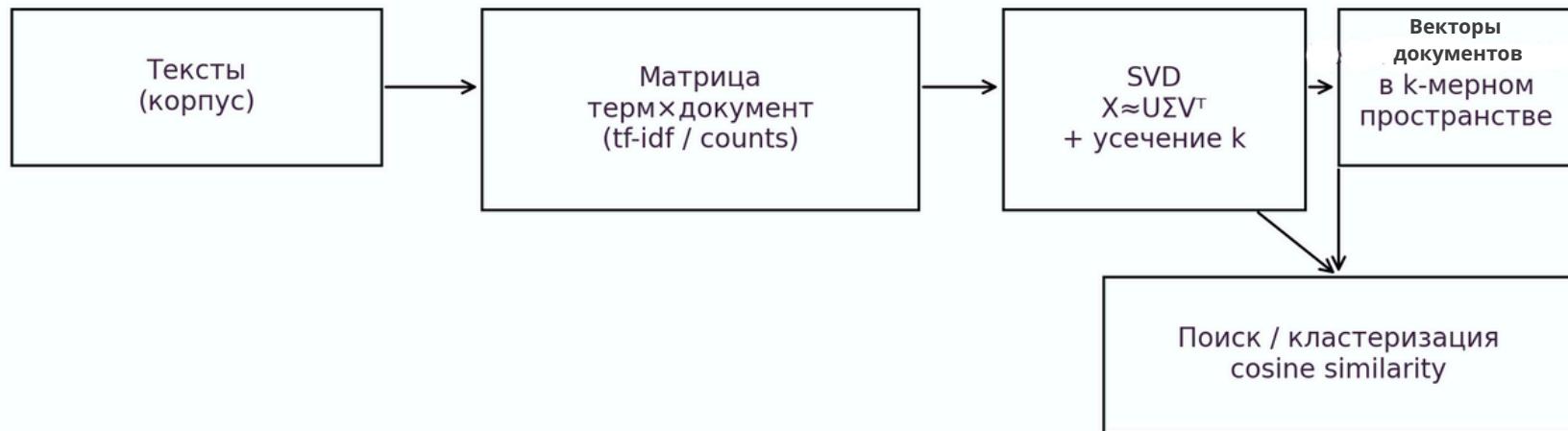
- частоты совместной встречаемости в окне $\pm k$ (или в документе),
- иногда вместо raw counts делают взвешивание.

Как превращается в вектора:

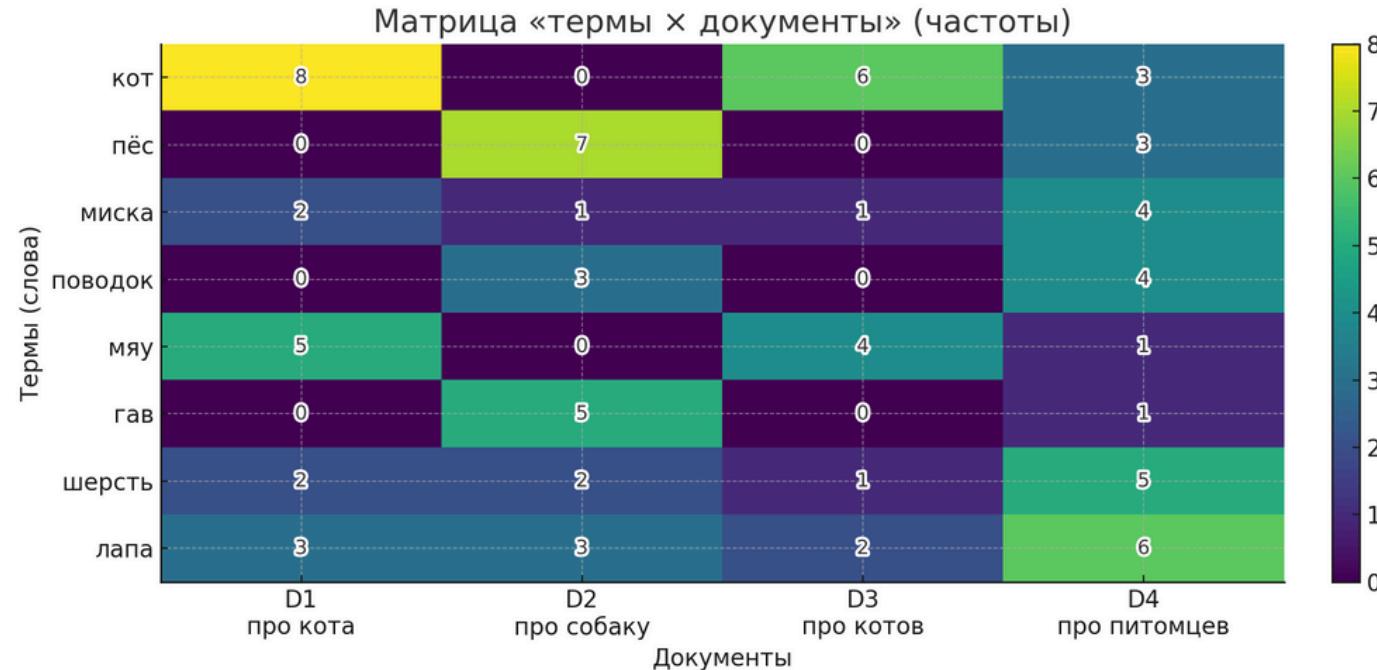
- **LSA/LSI**: TF-IDF матрица документов/термов → SVD → латентное пространство.
- **PPMI** + SVD: co-occurrence → PPMI → SVD → word vectors.
- **GloVe**: учит эмбеддинги так, чтобы хорошо аппроксимировать лог со-occurrence.



LSA/LSI: базовый пайплайн



1. строим матрицу “слова × документы” (частоты или TF-IDF)



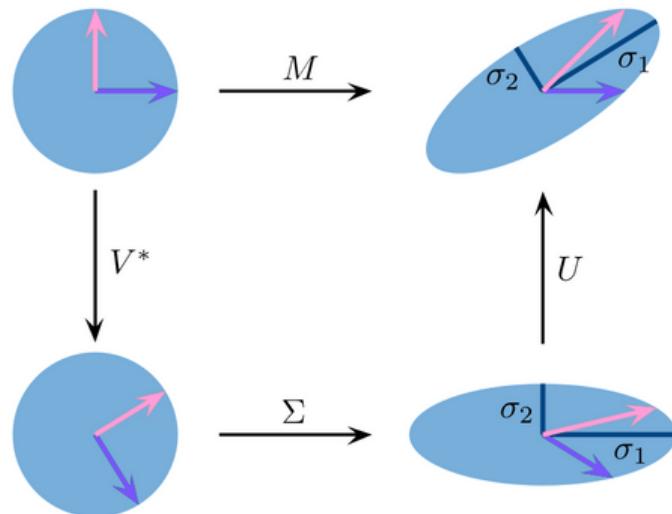
SVD (Singular Value Decomposition)

любая матрица - это “поворот + растяжение + поворот”

Роль матриц

- U - связь Слов с Темами
- Σ - Важность Тем
- V^* - связь Документов с Темами

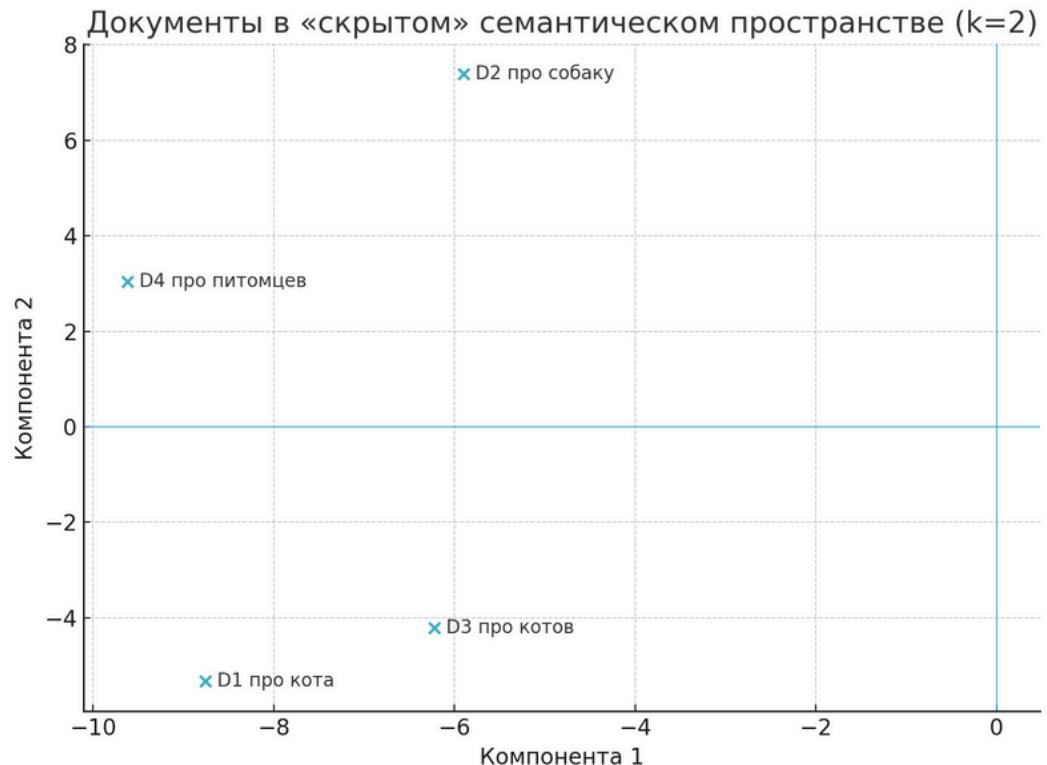
> если оставить только несколько крупнейших σ_i , получаем лучшую низкоранговую аппроксимацию - полезно для сжатия и “выделения главных скрытых факторов” в текстовой матрице



$$M = U \cdot \Sigma \cdot V^*$$

2. делаем SVD-разложение и оставляем только k главных компонент
3. сравниваем документы/ слова в низкоразмерном “семантическом” пространстве по косинусному сходству.

> Геометрия: документы (и слова) получают координаты в k-мерном пространстве; дальше - cosine similarity для поиска/кластеризации.





Latent Semantic Analysis

13

Место	Входные данные (Features)	Модель (Model)	Функция потерь (Loss)	Hitrate@3
11	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>cointegrated/rubert-tiny2</code>	Стандартная	0.2846
10	<code>app_name</code>	<code>FacebookAI/xlm-roberta-base</code>	Стандартная	0.5576
9	<code>app_name</code>	<code>FacebookAI/xlm-roberta-base</code>	С весами для классов	0.7017
8	<code>app_name</code> + <code>shortDescription</code>	<code>FacebookAI/xlm-roberta-base</code>	Стандартная	0.7714
7	<code>app_name</code> + <code>shortDescription</code>	<code>FacebookAI/xlm-roberta-base</code>	С весами для классов	0.8409
6	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>TF-IDF + LSA + Logistic Regression</code>	С весами для классов	0.8638
5	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>FacebookAI/xlm-roberta-base</code>	Стандартная	0.8941
4	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>TF-IDF + ClassifierChain(LR)</code>	С весами для классов	0.9092
3	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>FacebookAI/xlm-roberta-base</code>	С весами для классов	0.9165
2	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>intfloat/multilingual-e5-base</code>	С весами для классов	0.9196
1	<code>app_name</code> + <code>shortDescription</code> + <code>full_description</code>	<code>TF-IDF + Logistic Regression</code>	С весами для классов	0.9224



Positive Pointwise Mutual Information (PPMI)

14

Пусть у нас есть матрица частот $C(w,c)$ - сколько раз слово w встретилось рядом с контекстом c (обычно окно $\pm k$ слов).

1. Считаем суммы:

- $N = \sum_{w,c} C(w,c)$
- $C(w) = \sum_c C(w,c)$
- $C(c) = \sum_w C(w,c)$

2. Переводим в вероятности:

- $P(w, c) = \frac{C(w,c)}{N}$
- $P(w) = \frac{C(w)}{N}$
- $P(c) = \frac{C(c)}{N}$

3. PMI:

$$PMI(w, c) = \log \frac{P(w, c)}{P(w)P(c)}$$

4. PPMI:

$$PPMI(w, c) = \max(PMI(w, c), 0)$$

Интерпретация:
- $PMI > 0$: вместе чаще, чем "по случайности";
- $PMI < 0$: вместе реже, чем ожидали;
- PPMI оставляет только положительные ассоциации.

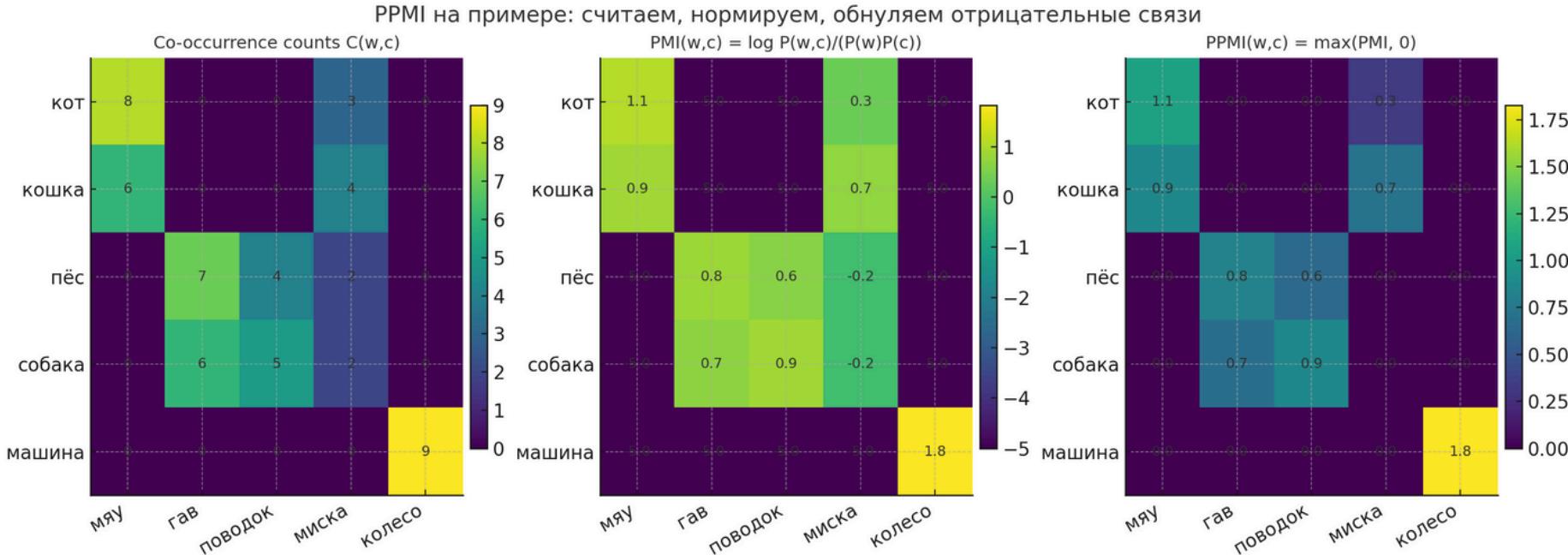
Практика:
PPMI часто подают на SVD → плотные эмбеддинги (LSA/LSI)

Pointwise Mutual Information (PMI)



Positive Pointwise Mutual Information (PPMI)

15





- PPMI = как взвешивать co-occurrence
- LSA/LSI = как сжать матрицу до низкого ранга через SVD

Интерпретируемая матрица “слово-контекст” → PPMI.

Компактное латентное пространство для документов/слов → LSA/LSI.

“Лучшее из двух миров” → PPMI → SVD (это классический count-based baseline)

Факторизация PPMI через SVD может давать качество сопоставимое с SGNS (Skip-gram with Negative Sampling) на задачах похожести слов).

[Neural Word Embedding as Implicit Matrix Factorization](#)

- Смысл слова проявляется в глобальной статистике совместных встречаемостей (co-occurrence).
- Особенно информативны отношения вероятностей контекстов: $P(k|i)/P(k|j)$ - в них “шум” от частых нерелевантных слов частично сокращается.

Матрица со-occurrence X_{ij} :
сколько раз слово j
встречается в окне
контекста слова i .

Что оптимизируем (сердце GloVe)

- Подбираем вектора слова w_i и контекста \tilde{w}_j так, чтобы их скалярное произведение приближало лог-ко-встречаемость:

$$J = \sum_{i,j} f(X_{ij}) \left(w_i^\top \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij} \right)^2$$

- Весовая функция, чтобы не переоценивать редкие и не “задавливаться” частыми:

$$f(x) = \begin{cases} (x/x_{\max})^\alpha, & x < x_{\max} \\ 1, & \text{иначе} \end{cases} \quad (\text{часто } x_{\max} = 100, \alpha = 3/4)$$

GloVe: Global Vectors for Word Representation

Лог-билинейная регрессия / взвешенная матричная факторизация, где мы учим два набора векторов (для слова и для контекста) и биасы так, чтобы их скалярное произведение приближало \log co-occurrence.

В GloVe модель по сути считает **скор**:

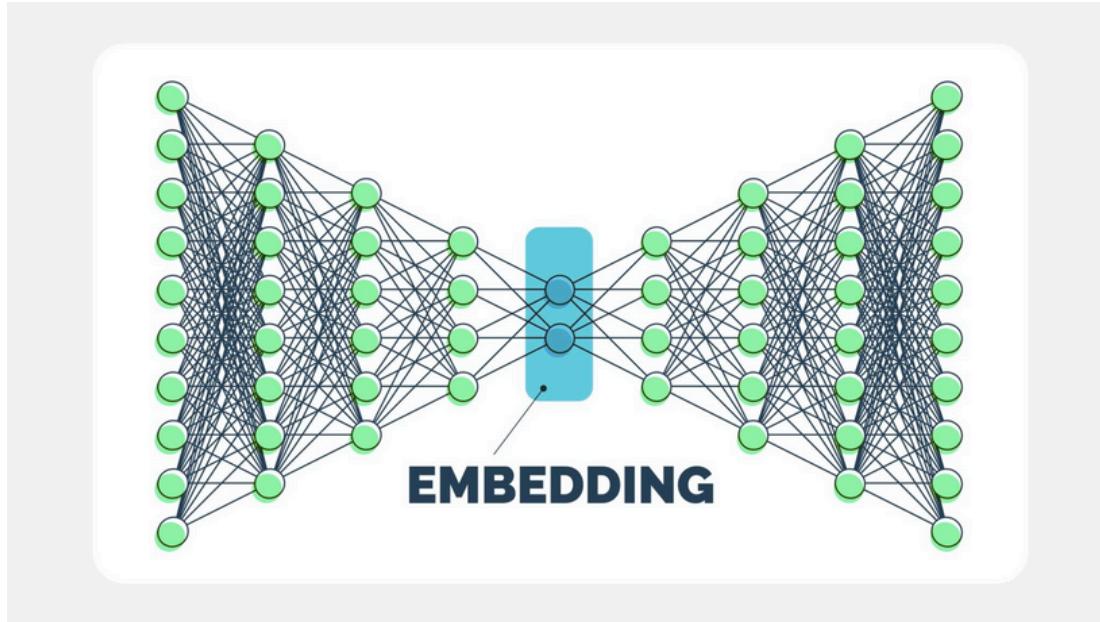
$$s(i, j) = w_i^T \tilde{w}_j + b_i + \tilde{b}_j$$

и подгоняет его к $\log X_{ij}$ (взвешенным MSE по ненулевым X_{ij}).

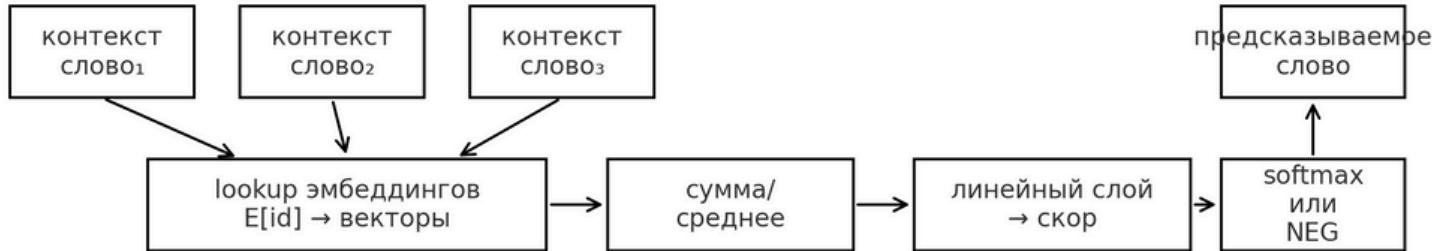


Нейросетевые эмбеддинги: Word2Vec

19



Word2Vec: a Prediction-Based Method

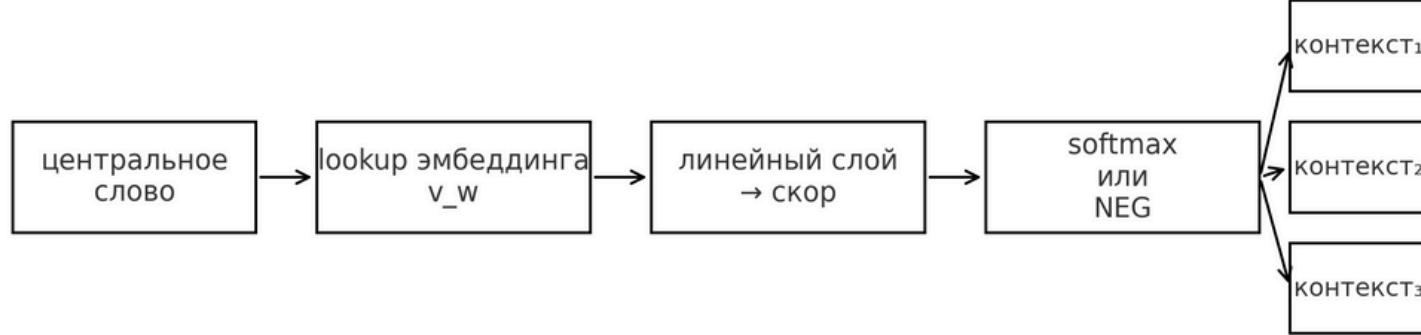


Идея: по словам вокруг восстановить центральное слово (порядок слов в окне игнорируется).

Логика: “Заполнить пробел”. Мы скармливаем сети контекст (слова вокруг) и просим угадать слово в центре.

- Вход: [Кот, на, коврике]
- Задача: Угадай скрытое слово.
- Ответ: сидит.

Особенность: Быстро тренируется, сглаживает векторы (усредняет контекст). Хорош для частотных слов.



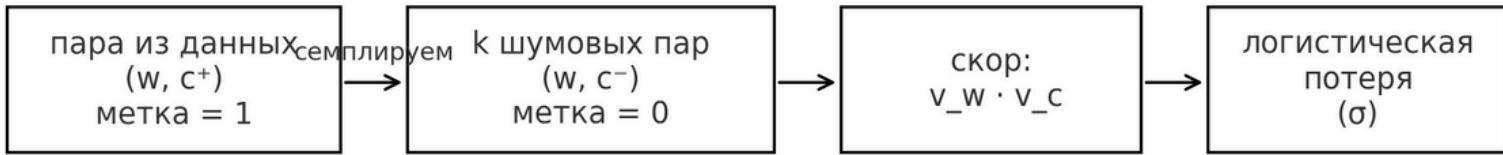
Идея: по центральному слову предсказывать слова вокруг (каждое (w, c) — отдельный пример).

Логика: “Угадай соседей”. Мы даем сети слово из центра и просим предсказать, какие слова могут стоять рядом с ним.

- Вход: [сидит]
- Задача: Угадай контекст.
- Ответ: [Кот, на, коврике].

Особенность: Работает медленнее, но гораздо лучше понимает редкие слова и тонкие смысловые связи. Сегодня чаще используют именно его.

Negative Sampling (NEG): интуиция



Смысл:

- делаем $v_w \cdot v_{c^+}$ большим (позитивные пары)
- делаем $v_w \cdot v_{c^-}$ маленьким (негативные пары)
- вместо полного softmax по словарю обучаемся на нескольких «контрастных» примерах

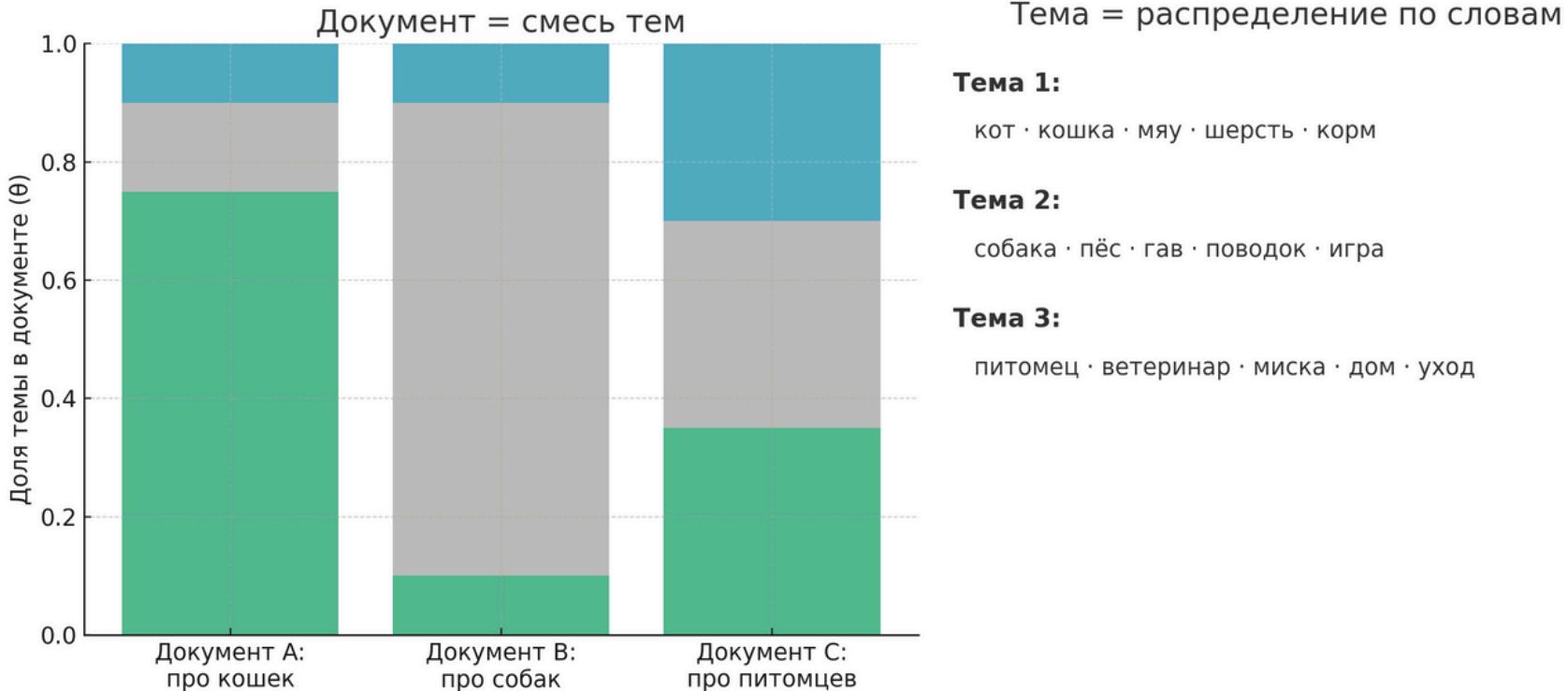
Мы применяем трюк. Для каждой реальной пары (сидит, коврике) мы берем 5-10 случайных слов из словаря (“апельсин”, “демократия”, “бежать”) и говорим сети:

- Сделай вектор “сидит” близким к вектору “коврике”.
- Сделай вектор “сидит” далеким от векторов “апельсин”, “демократия”.



Идеи тематического моделирования

23





LDA - Latent Dirichlet Allocation

(вероятностный подход)

24

Документ = смесь тем: θ_d — распределение тем в документе.

Тема = распределение слов: ϕ_k — вероятности слов в теме.

Модель объясняет, как “могли быть сгенерированы” слова: выбираем тему z из θ_d , затем слово w из ϕ_z .

Важное практическое отличие: всё неотрицательно и суммы = 1 \Rightarrow темы часто проще показывать как “топ-слова + вероятности”.

Latent Dirichlet Allocation

