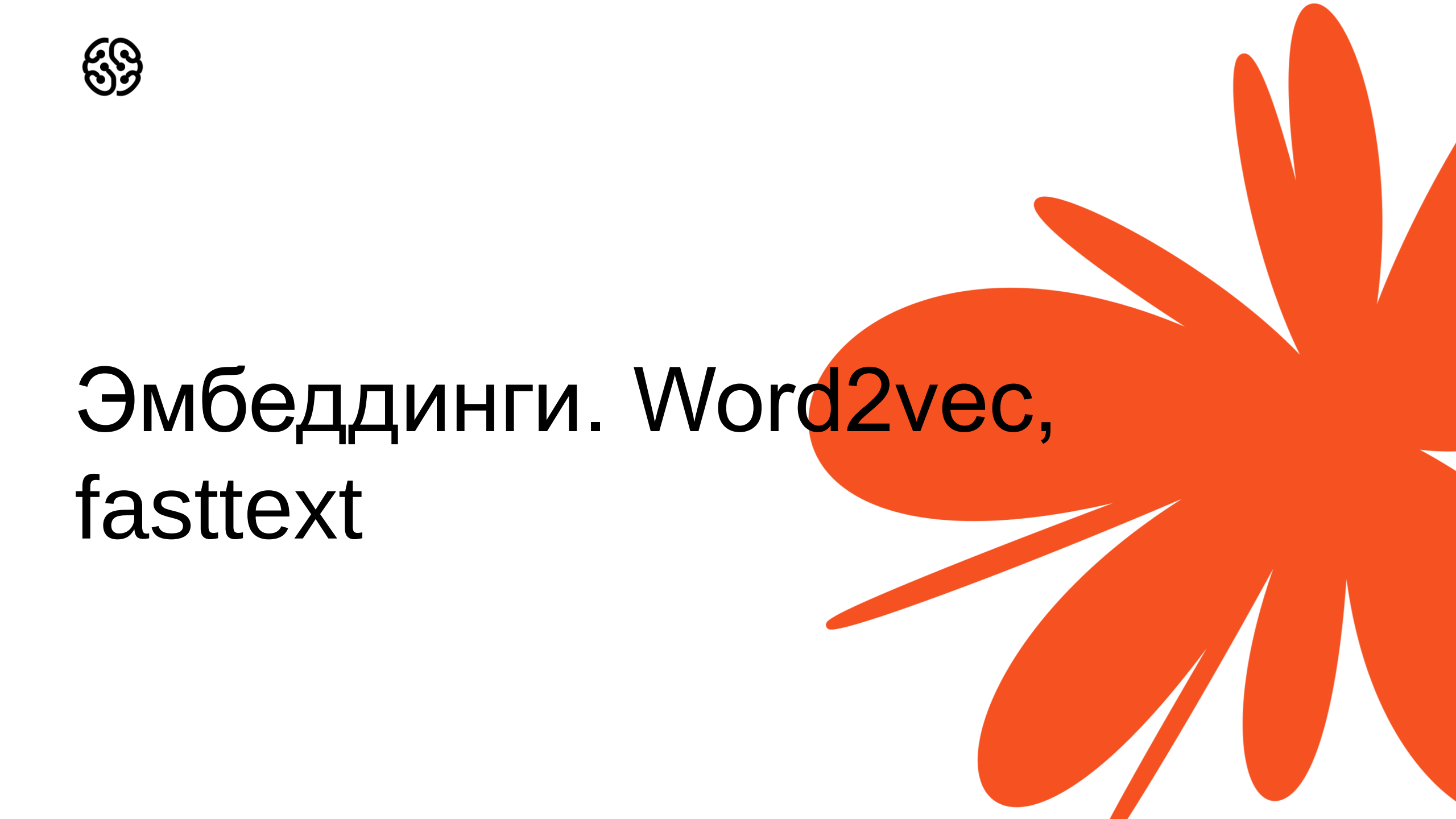


Эмбеддинги. Word2vec, fasttext

Представления слов

Слова представляются в виде one-hot вектора:

$$\begin{aligned} \text{умный} &\quad [0, \ldots, 1, \ldots, 0, \ldots, 0, \ldots, 0] \\ \text{сообразительный} &\quad [0, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 0] \\ \text{собака} &\quad [0, \ldots, 0, \ldots, 0, \ldots, 1, \ldots, 0] \end{aligned}$$

Недостатки:

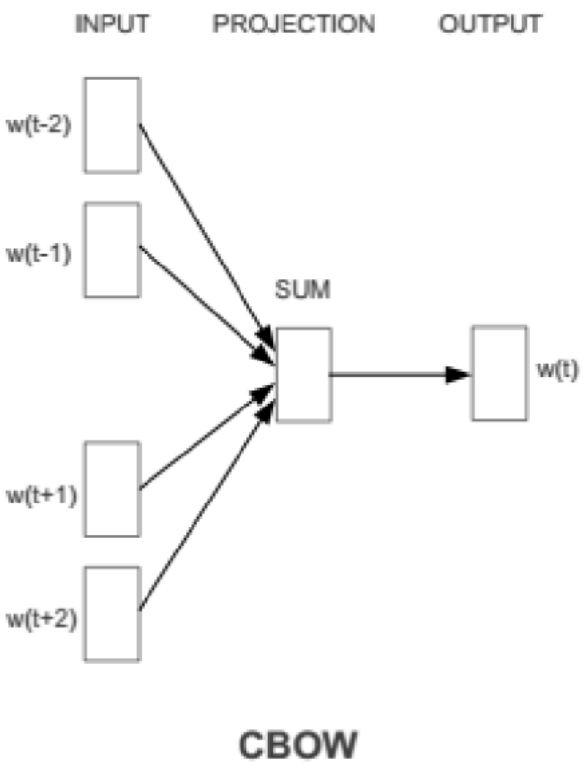
- Большая размерность векторов;
- Все вектора одинаково похожи;
- Сложно кодировать дополнительную информацию.

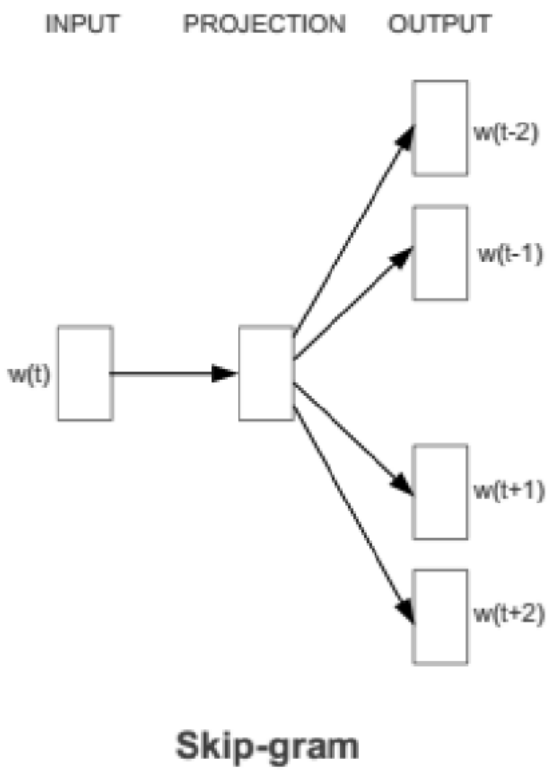
Гипотеза компактности

Близкие (похожие) слова встречаются в похожем контексте.

Идея: у близких слов (синонимов, родственных слов) и представления получаются близкие.

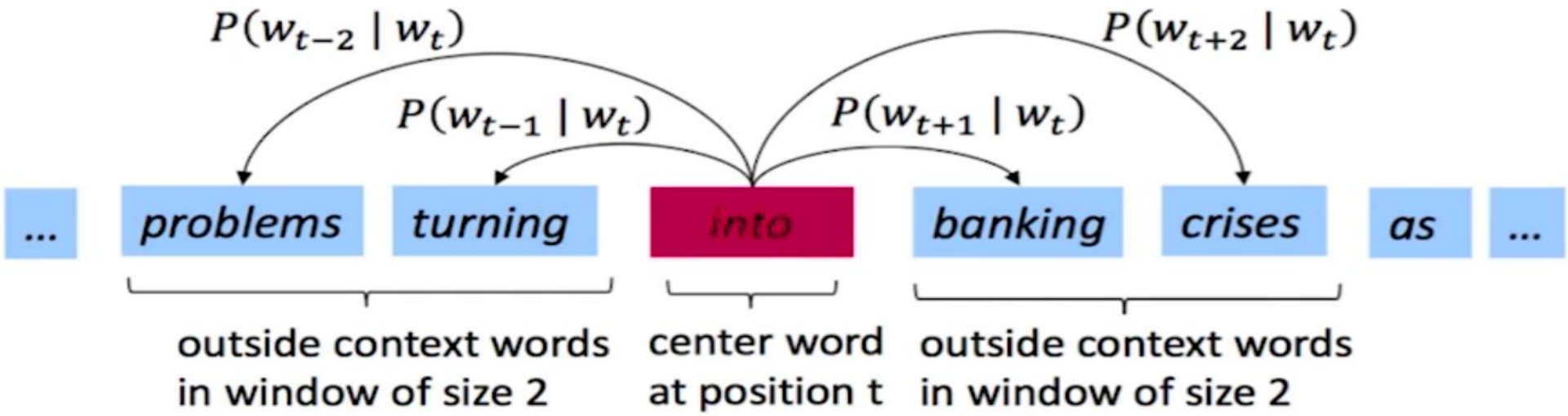
Word2Vec


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
CBOW


INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Skip-gram

Skip-gram

Зная контекст, мы хотим максимизировать вероятность центрального слова. Перемещаемся скользящим окном.



Skip-gram

Вероятность встретить слово w_0 рядом со словом w_1:

$$p(w_O|w_I) = \frac{\exp(\langle v'_{w_O}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

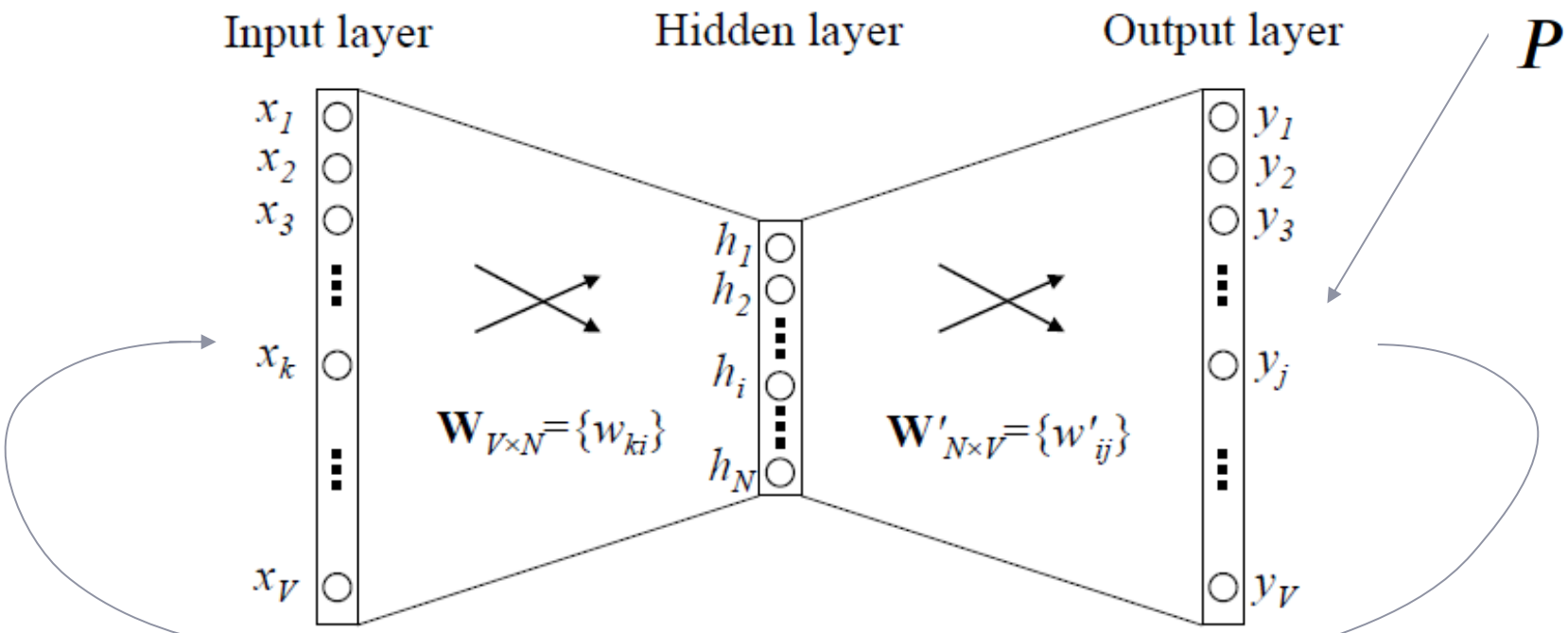
Функционал для текста $T=(w_1 w_2 \dots w_n)$:

$$\sum_{i=1}^{n} \sum_{\substack{-c \le j \le c \\ j \neq 0}} \log p(w_{i+j}|w_i) \to \max$$

Skip-gram

SoftMax models

$$P(w_{t+1} \mid w_t; \theta)$$



... problems turning **into** banking crises as ...

Skip-gram

Простейшая модель классификации — линейный классификатор, применяемый к вектору контекста.

$$softmax(\boldsymbol{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} = p_i$$

Для обучения минимизируем функцию потерь.

Skip-gram

Вероятность встретить слово w_0 рядом со словом w_1:

$$p(w_O|w_I) = \frac{\exp(\langle v'_{w_O}, v_{w_I} \rangle)}{\sum_{w \in W} \exp(\langle v'_w, v_{w_I} \rangle)}$$

Считать знаменатель *крайне* затратно. Значит, и производные считать тоже долго.

Negative sampling

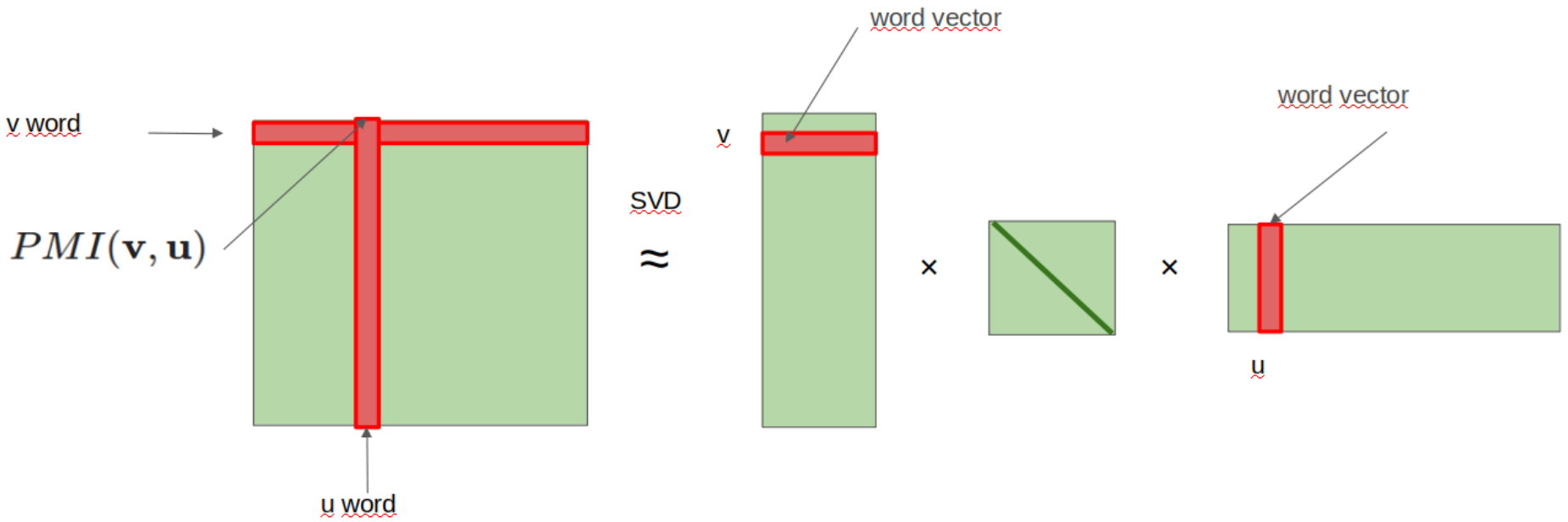
$$p(w_O|w_I) = \log \sigma(\langle v'_{w_O}, v_{w_I} \rangle) + \sum_{i=1}^{k} \log \sigma(-\langle v'_{w_i}, v_{w_I} \rangle)$$

w_i — случайно выбранные слова. Слово w генерируется с вероятностью $P(w)$ — шумовое распределение.

$P(w) = \frac{U(w)^{\frac{3}{4}}}{\sum_{v \in W} U(v)^{\frac{3}{4}}}$, $U(v)$ — частота слова v в корпусе текстов.

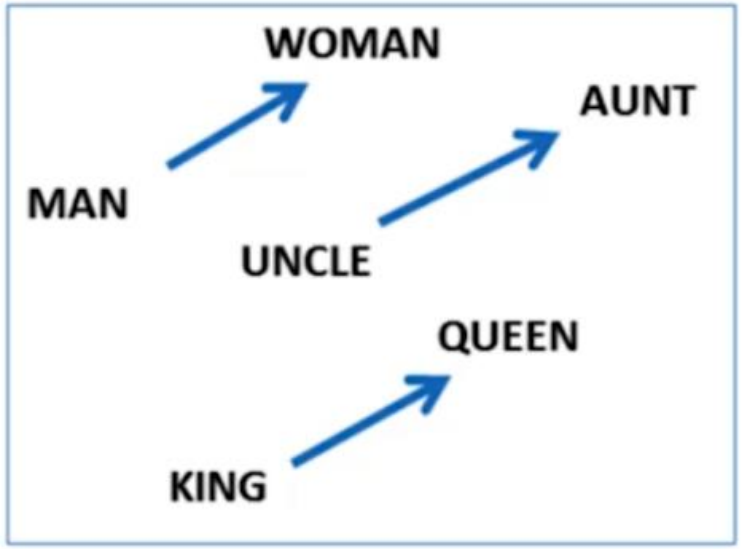
Word2vec vs SVD

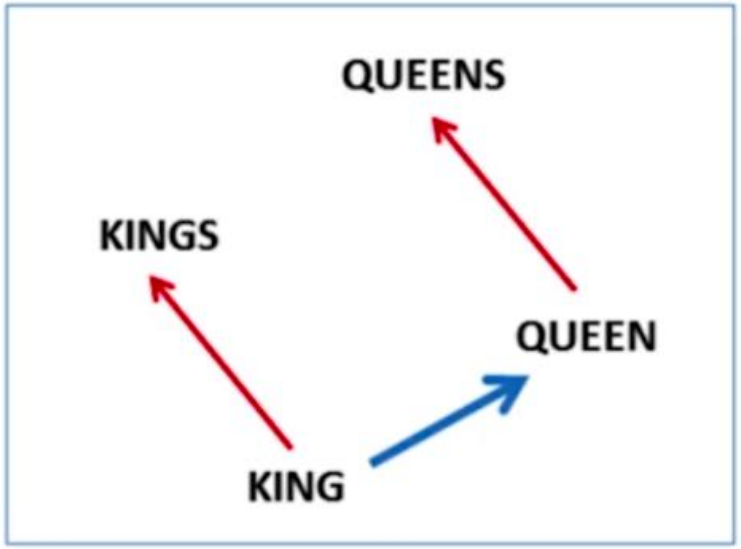
Word2Vec with negative sampling ≈ matrix factorization



Word2vec

v(king) - v(man) + v(woman) ≈ v(queen)





Fasttext

Разделите слово на мешок из *n*-граммов: *apple = <ap, ppl, ple, le> (BPE).*

Вычислите вектор для каждого *n*-грамма.

Вектор для слова = сумма <вектора слова, векторов для n-граммов слов>.

Ваши вопросы?

Классические подходы к обработке естественного языка, урок 3