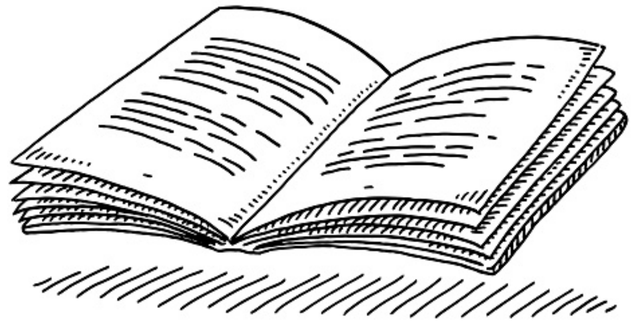
Работа с текстами

Елена Кантонистова

elena.kantonistova@yandex.ru
ekantonistova@hse.ru



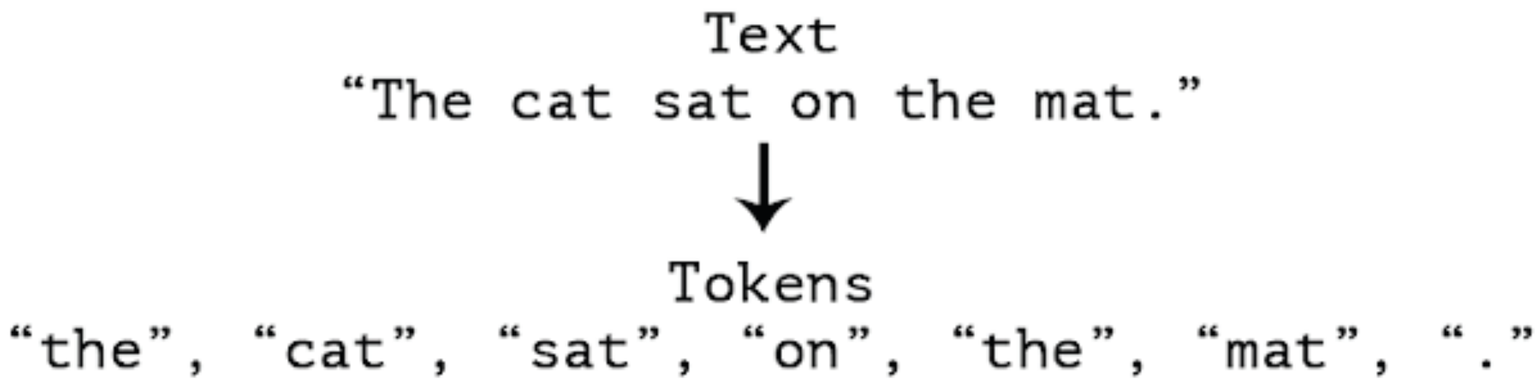
ВШЭ 2023

Терминология

- *Документ* - текст
- *Корпус* - набор документов
- *Токен* – формальное определение “слова”; токен может не иметь смыслового значения (например, “12fdh” или “авыдшл”), но обычно отделен от остальных токенов пробелами или знаками препинания

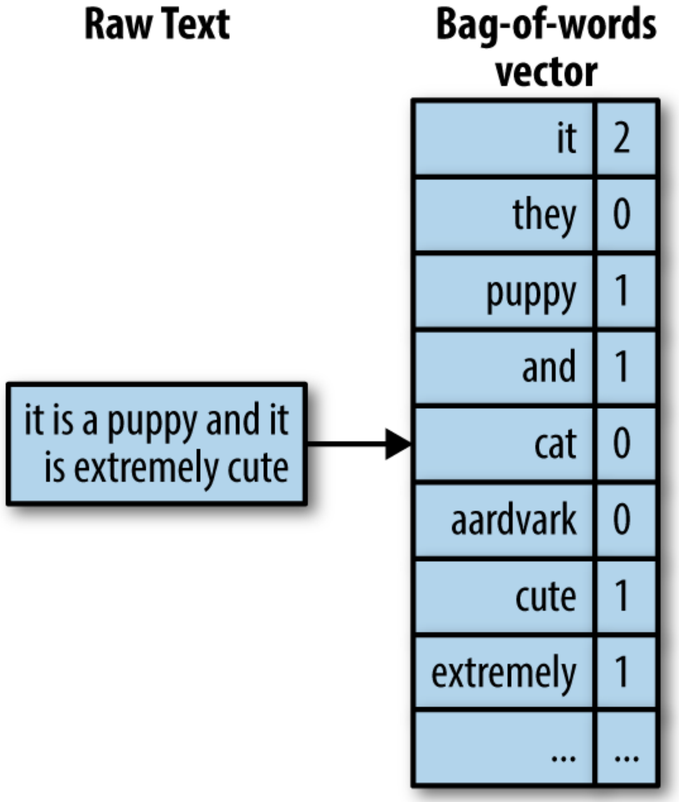
Токенизация текста

Чтобы работать с текстом, необходимо разбить его на токены. В простейшем случае токены – это слова (а также наборы букв, знаки препинания и т.д.).



Bag of words (мешок слов)

- По корпусу создадим словарь из всех встречающихся в нем слов (можно убрать общеупотребительные часто встречающиеся слова и очень редкие слова).
- Каждое слово закодируем вектором, в котором стоит единица на месте, соответствующем месту этого слова в словаре, все остальные компоненты вектора – 0.
- Для кодирования документа сложим коды всех его слов.



Bag of words (пример)

Пусть корпус состоит из следующих документов:

- D1 - “I am feeling very happy today”
- D2 - “I am not well today”
- D3 - “I wish I could go to play”

Кодировка этих документов будет такой:

	I	am	feeling	very	happy	today	not	well	wish	could	go	to	play
D1	1	1	1	1	1	1	0	0	0	0	0	0	0
D2	1	1	0	0	0	1	1	1	0	0	0	0	0
D3	2	0	0	0	0	0	0	0	1	1	1	1	1

Bag of words

Используя bag of words (BOW), мы теряем информацию о порядке слов в документе.

Пример: векторы документов “**I have no cats**” и “**No, I have cats**” будут идентичны.

Tf-idf

- Слова, которые редко встречаются в корпусе, но присутствуют в документе, могут оказаться важными для характеристики документа
- Слова, которые встречаются во всех документах, наоборот, не важны.

Tf-idf

Tf-idf слова $\boldsymbol{t}$ в документе $\boldsymbol{d}$ из корпуса $\boldsymbol{D}$:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

- $tf(t, d)$ - частота вхождения слова t в документ d
- $idf(t, D)$ - величина, обратная частоте, с которой слово t встречается в корпусе D (обычно от нее еще берут логарифм)

Tf-idf

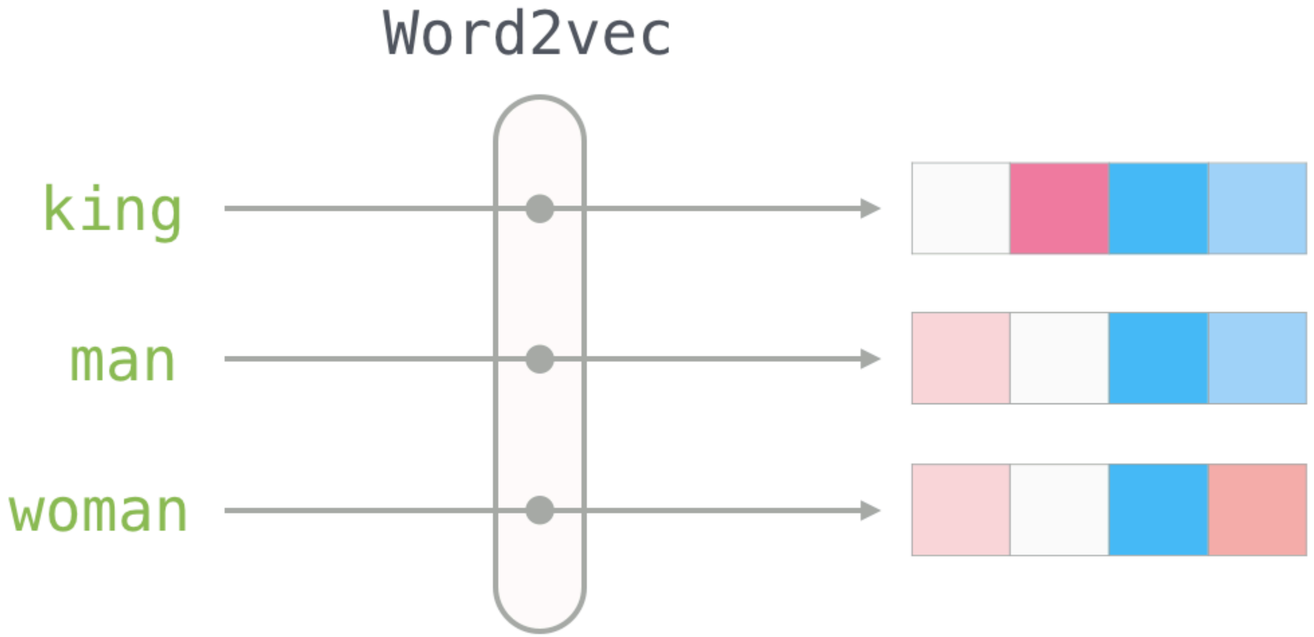
D1: He is a lazy boy. She is also lazy.

D2: Neeraj is a lazy person.

	He	She	lazy	boy	Neeraj	person
D1	0.06	0.06	0	0.06	0	0
D2	0	0	0	0	0.1	0.1

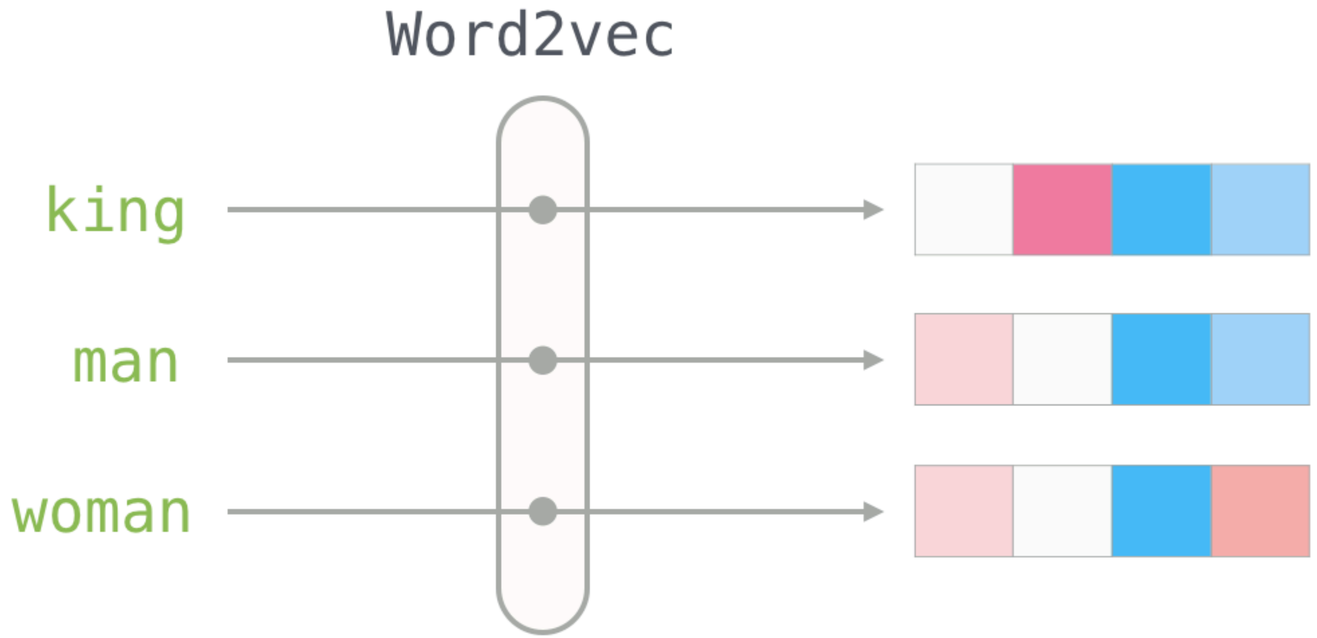
Word2Vec

Цель: для каждого слова из текста получить такой числовой вектор, чтобы векторы похожих по смыслу слов были “близки”.



Word2Vec

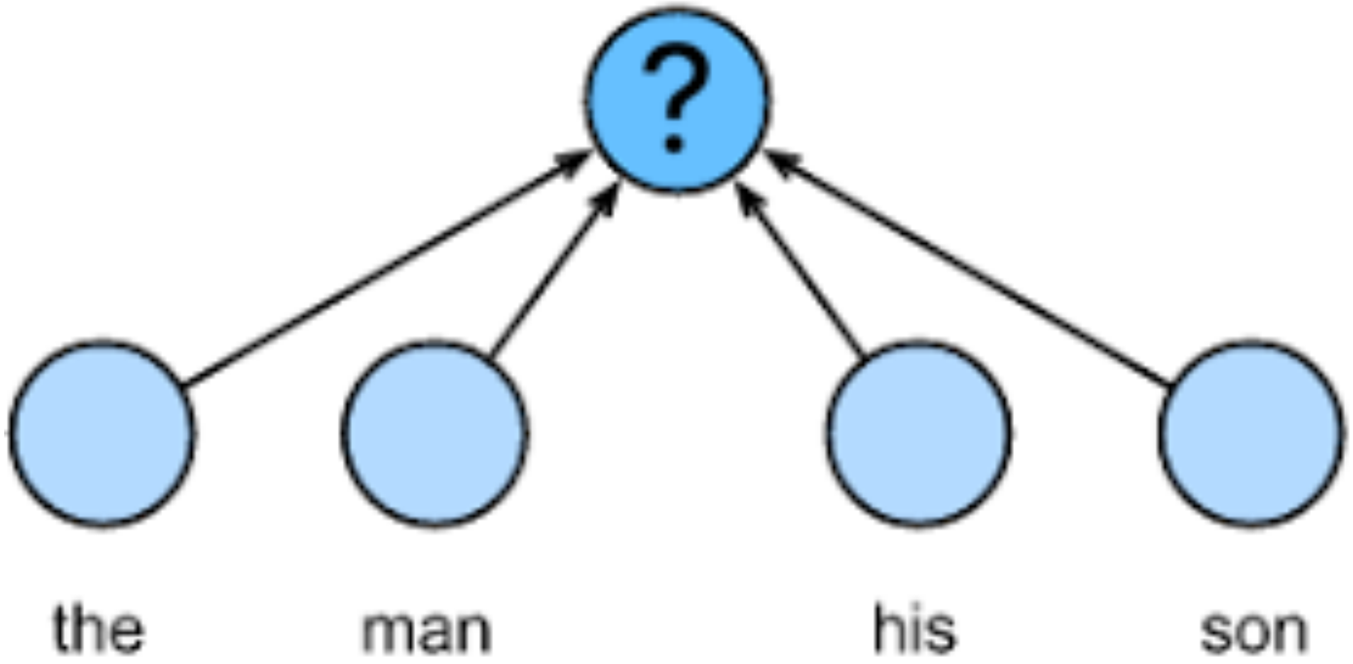
Цель: для каждого слова из текста получить такой числовой вектор, чтобы векторы похожих по смыслу слов были “близки”.



- В 2013 году Томас Миколов и его коллеги предложили word2vec – нейронную сеть, которую можно быстро обучить на огромном объеме текстов для получения векторов слов.

Вспомогательная задача

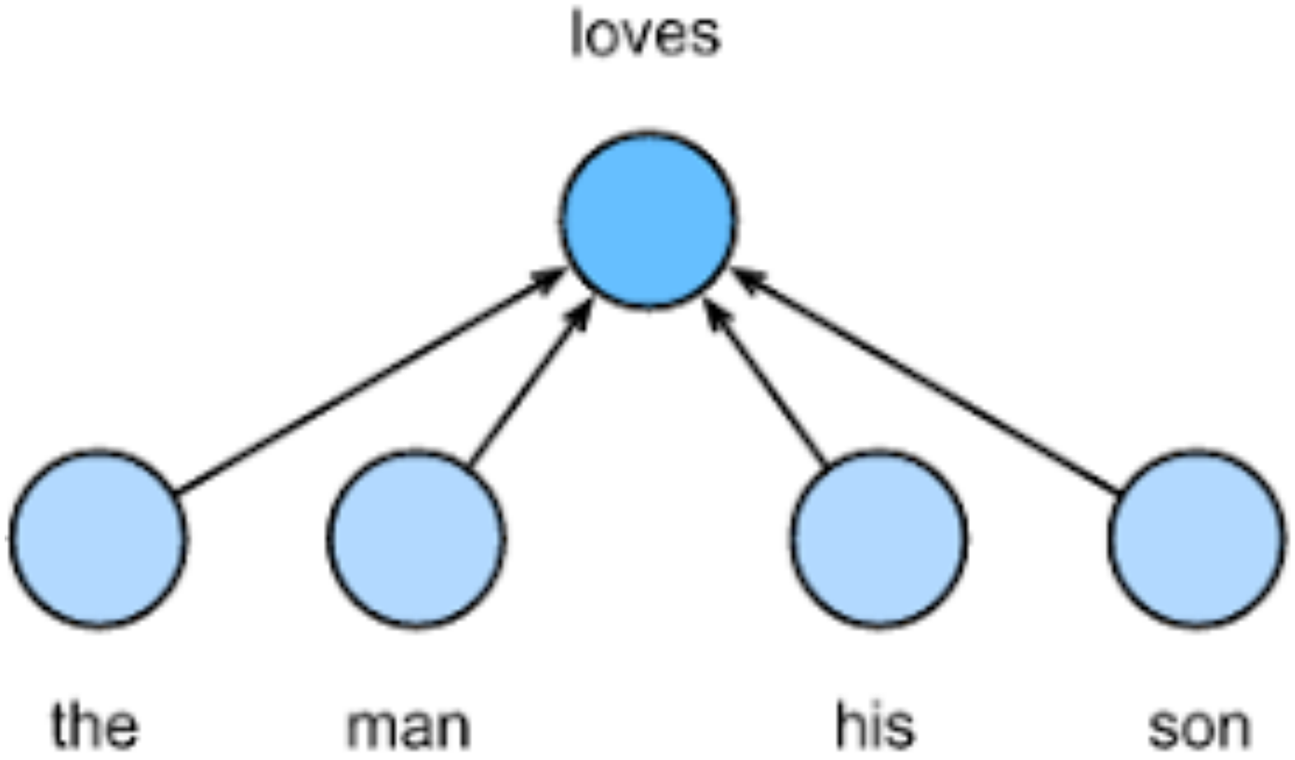
Задача: по контексту хотим предсказать какое слово стоит внутри



Вспомогательная задача

Ответ может быть таким:

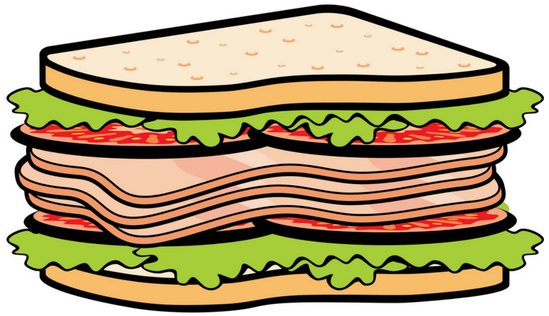
- loves - 0.7
- needs - 0.25
- holds - 0.05



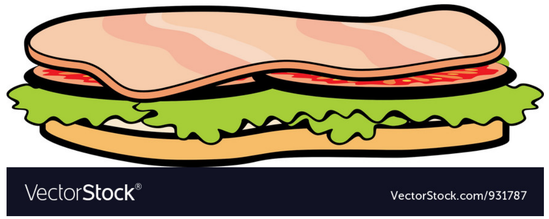
Почему именно такая задача?

Мы работаем в предположении, что **слова, встречающиеся в похожих контекстах, похожи!**

Например, слова *бутерброд* и *сэндвич* часто встречаются в одинаковых контекстах - значит, модель присвоит им похожие векторы.



$$s = (0.51,0.7,0.82,...)$$

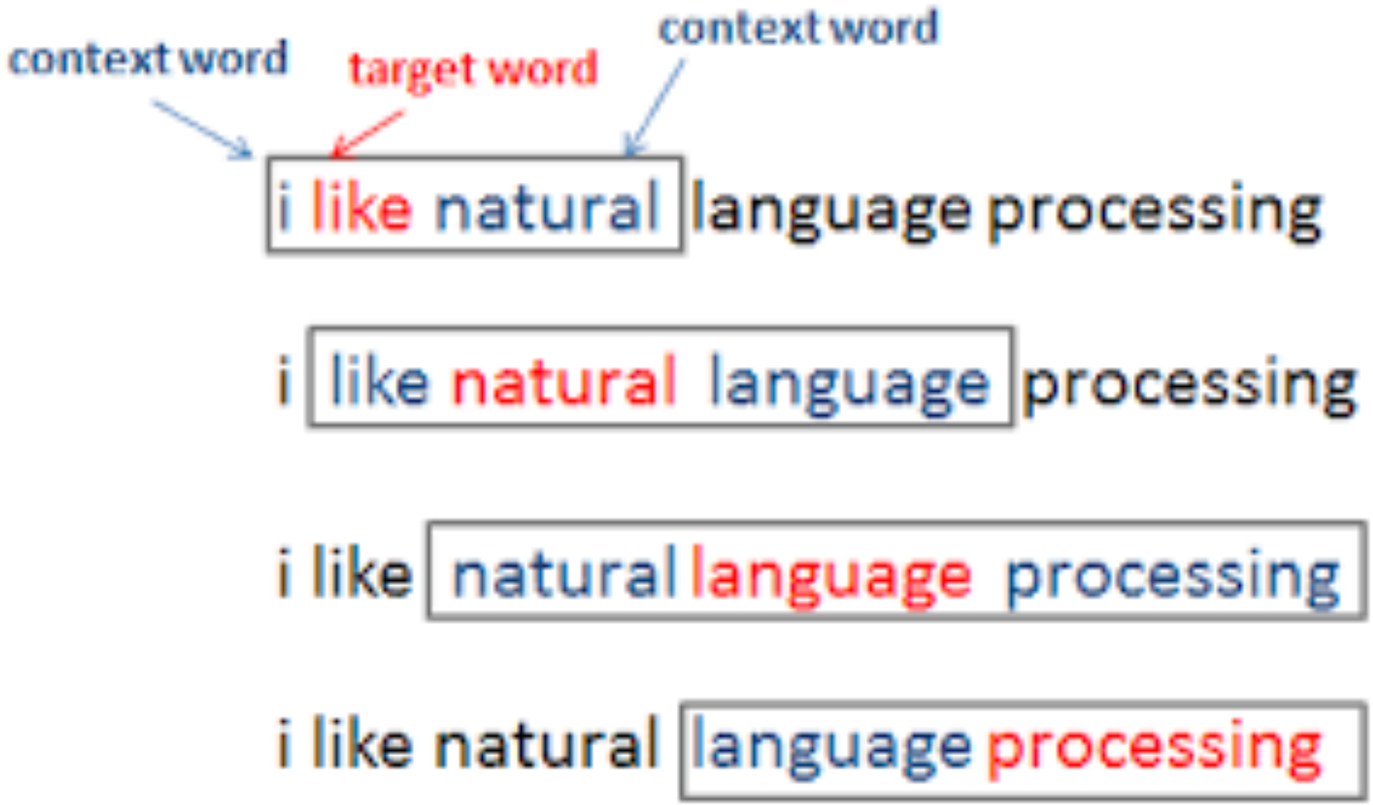


$$b = (0.45,0.72,0.83,...)$$

Где взять данные для обучения?

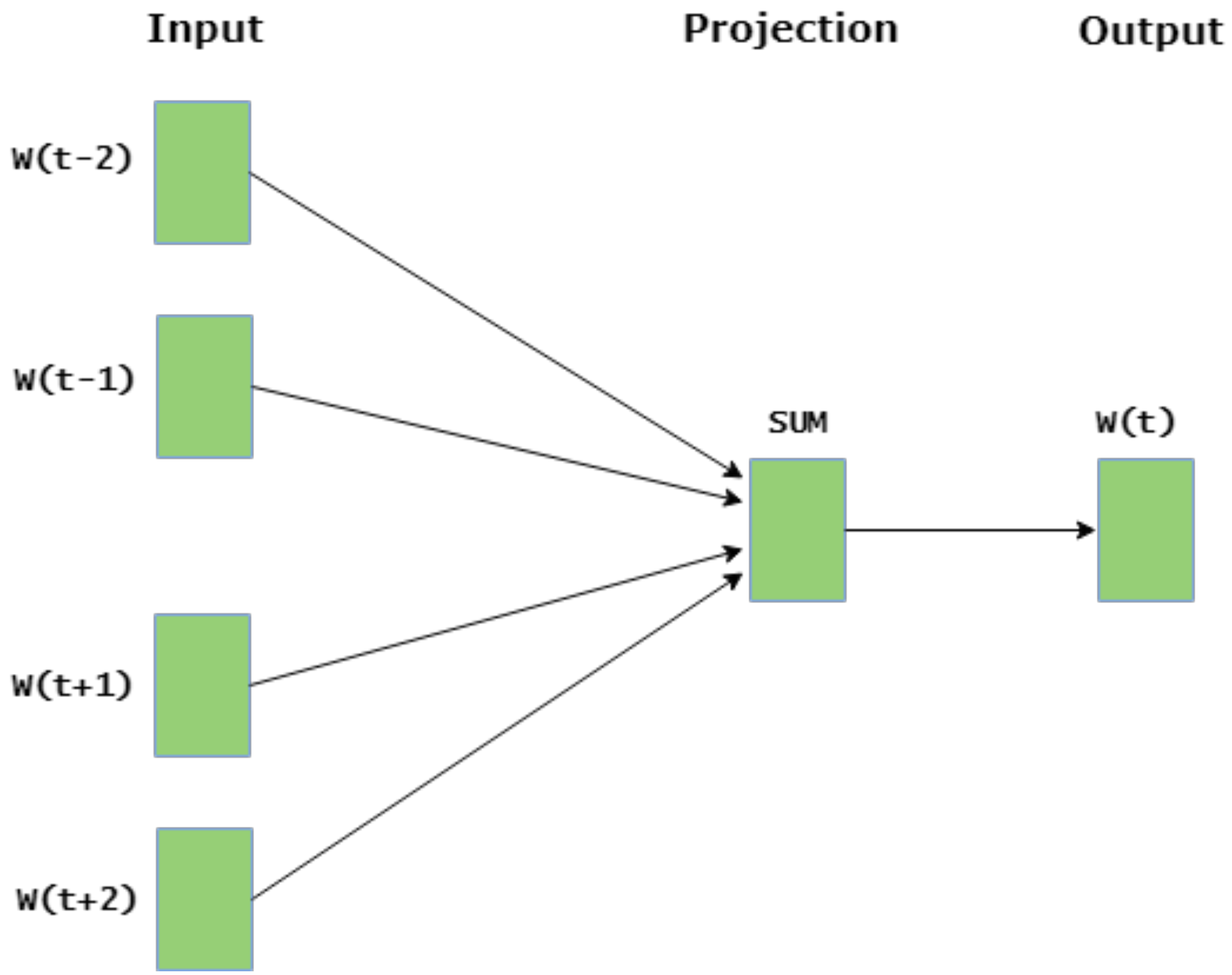
С помощью скользящего окна движемся по тексту:

- объекты - контекст (окружение центрального слова в окне)
- ответы - центральное слово



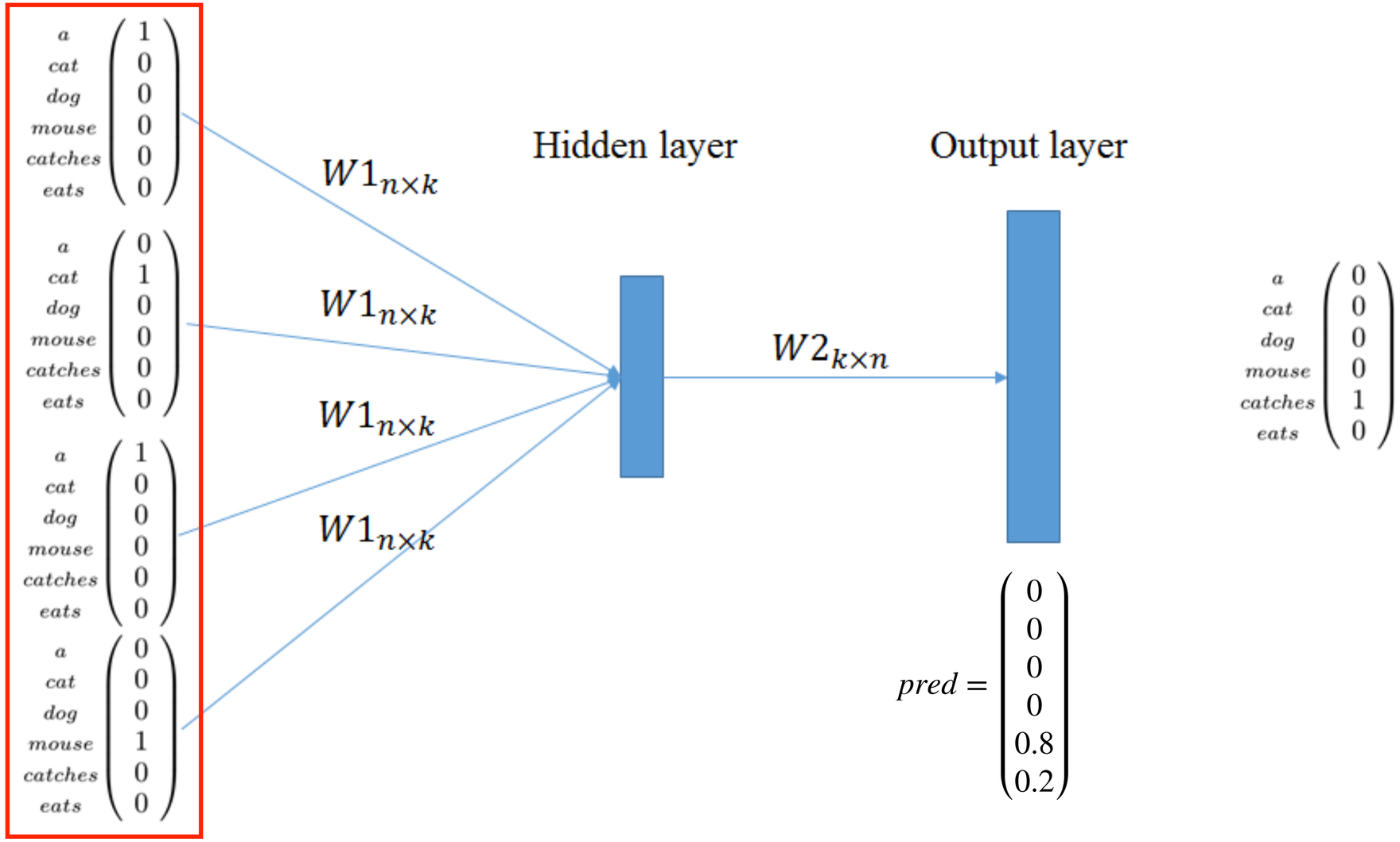
Как устроен алгоритм?

Общая схема такая:



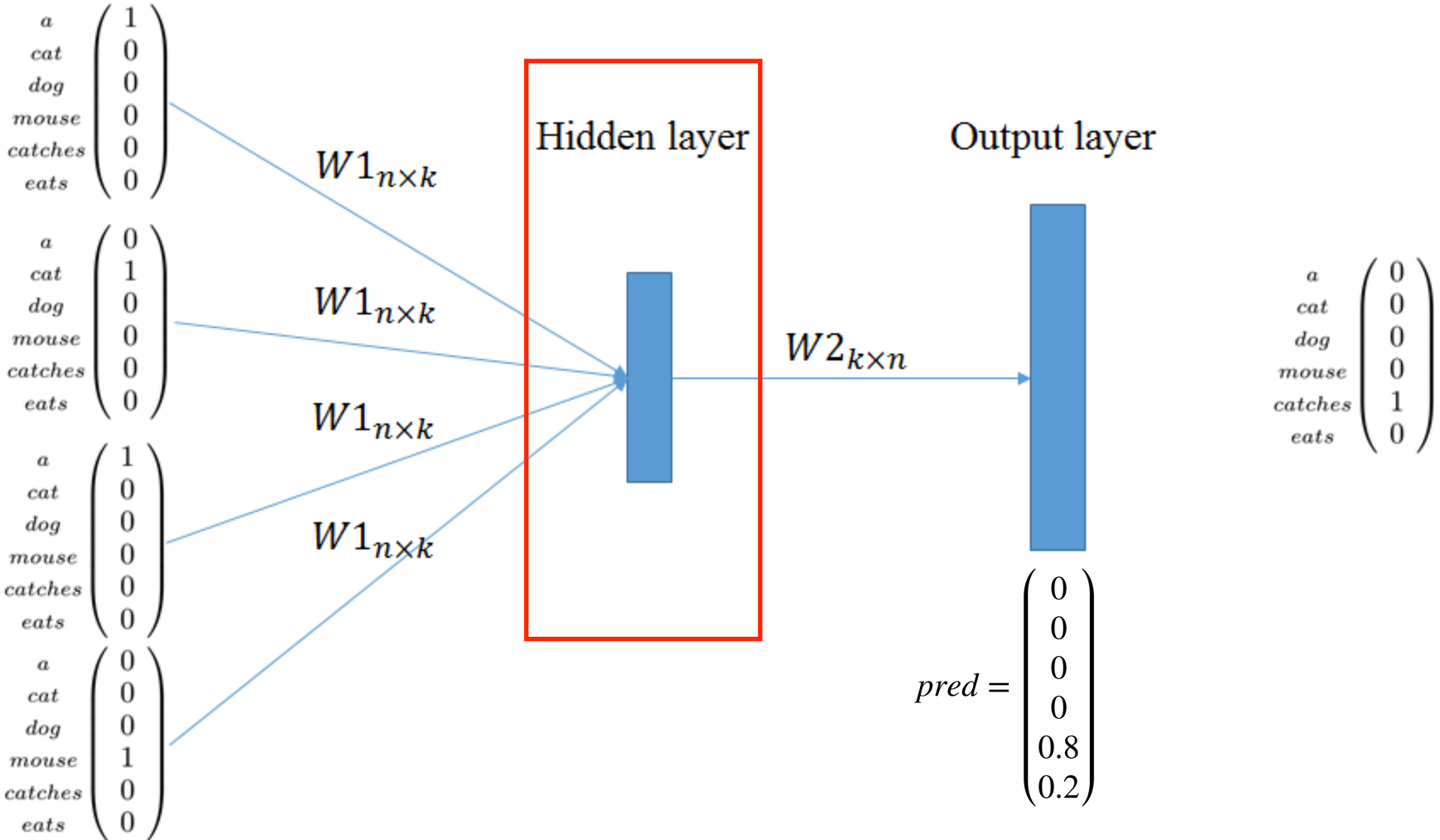
Как устроен алгоритм?

Input layer: каждое слово подаётся в виде OneHot-вектора.



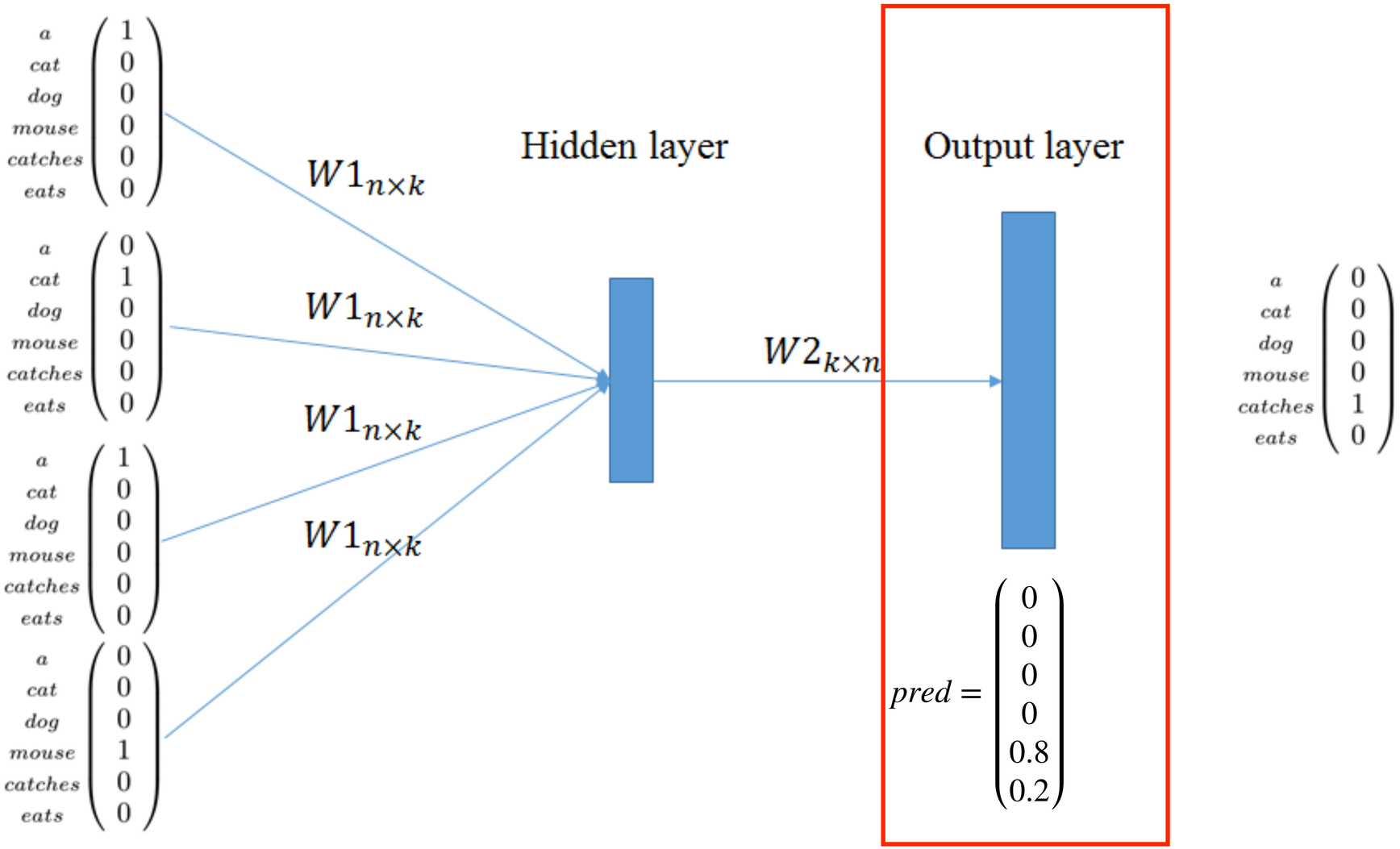
Как устроен алгоритм?

Hidden layer: полносвязный слой БЕЗ функции активации (то есть просто сумма векторов входных слов с весами).



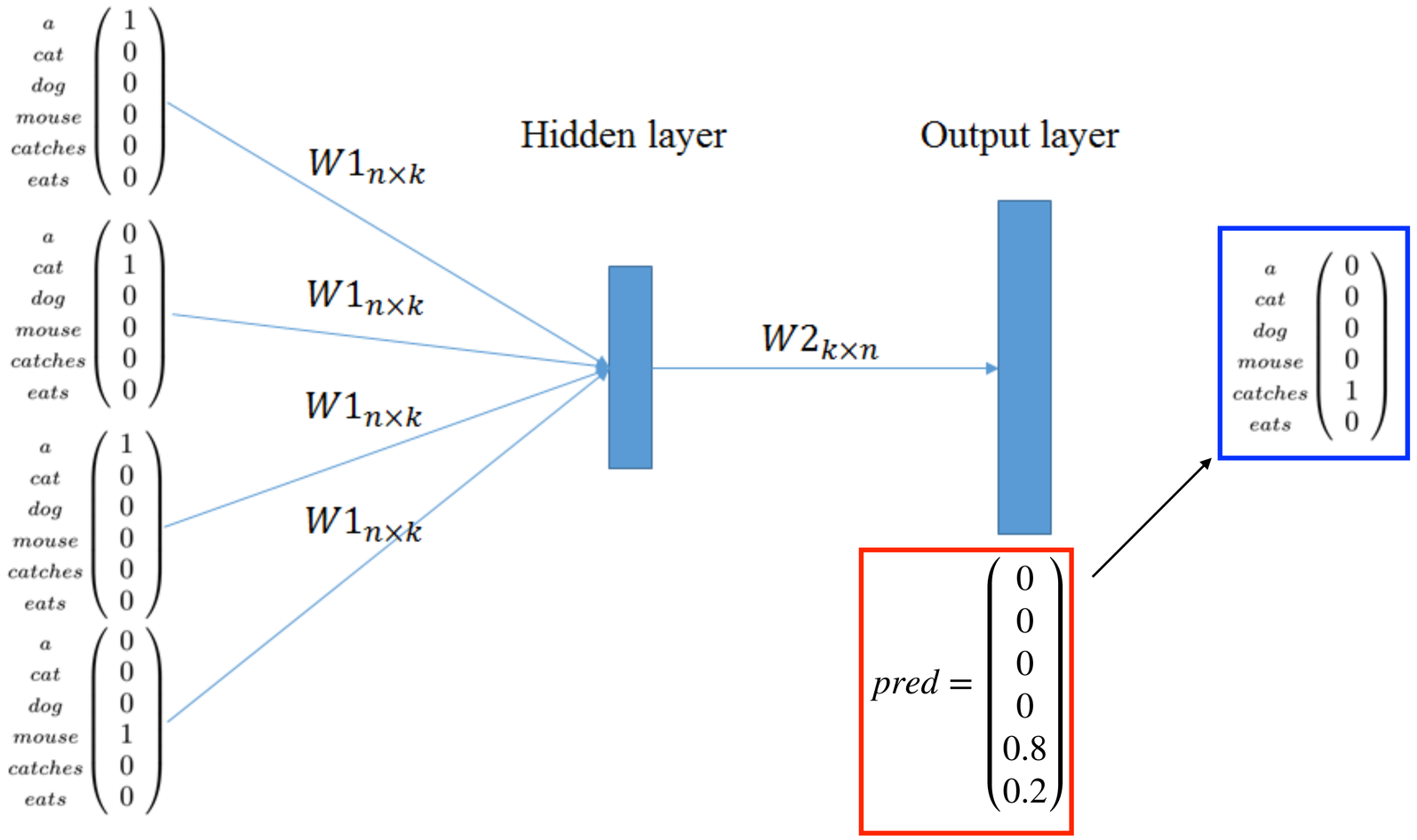
Как устроен алгоритм?

Output layer: вектор вероятностей, предсказанный моделью.



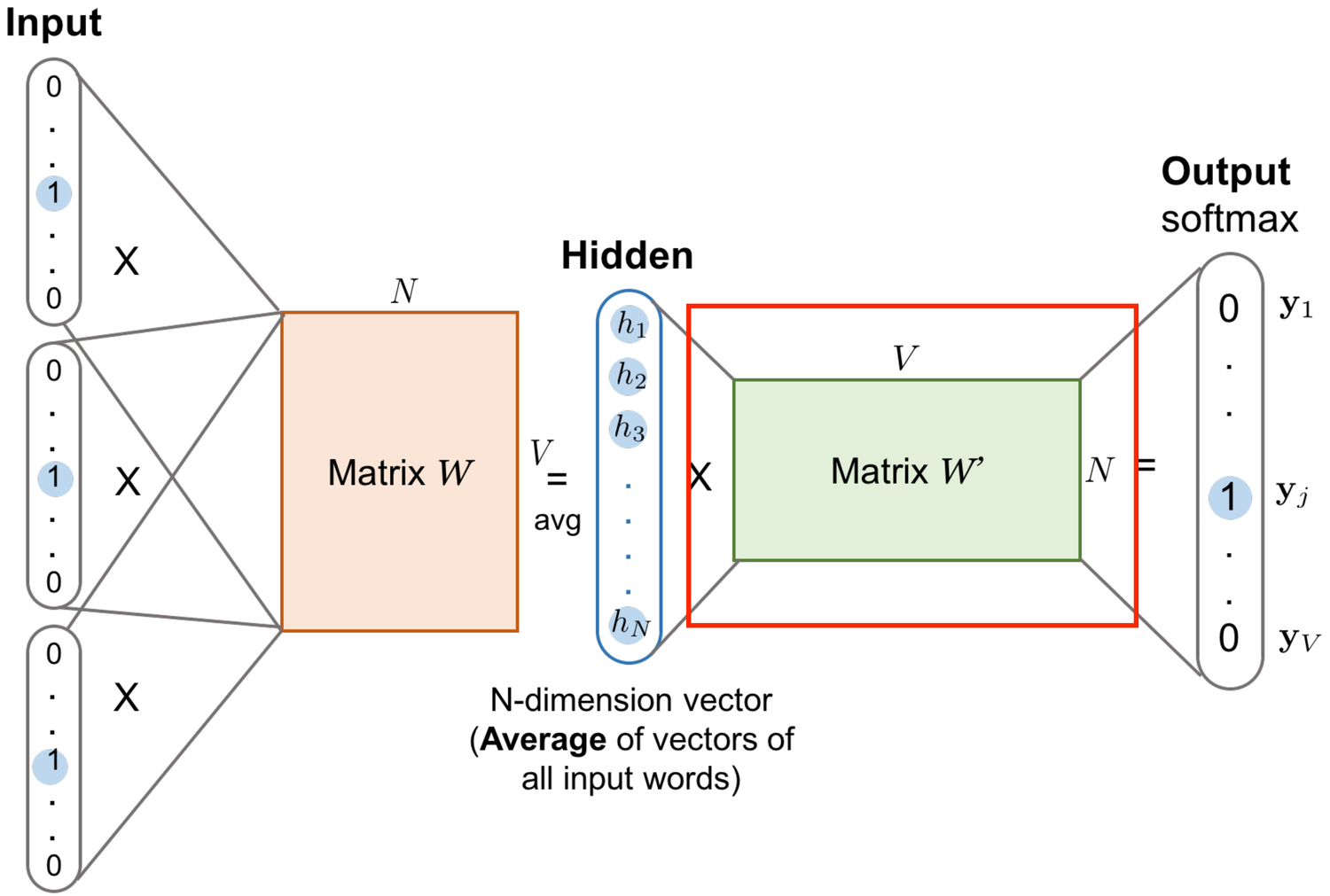
Какая функция потерь?

Функция потерь: Cross-Entropy Loss (Log-Loss) - используется во всех задачах классификации, где предсказываем вероятности классов



Помните, что мы ищем?

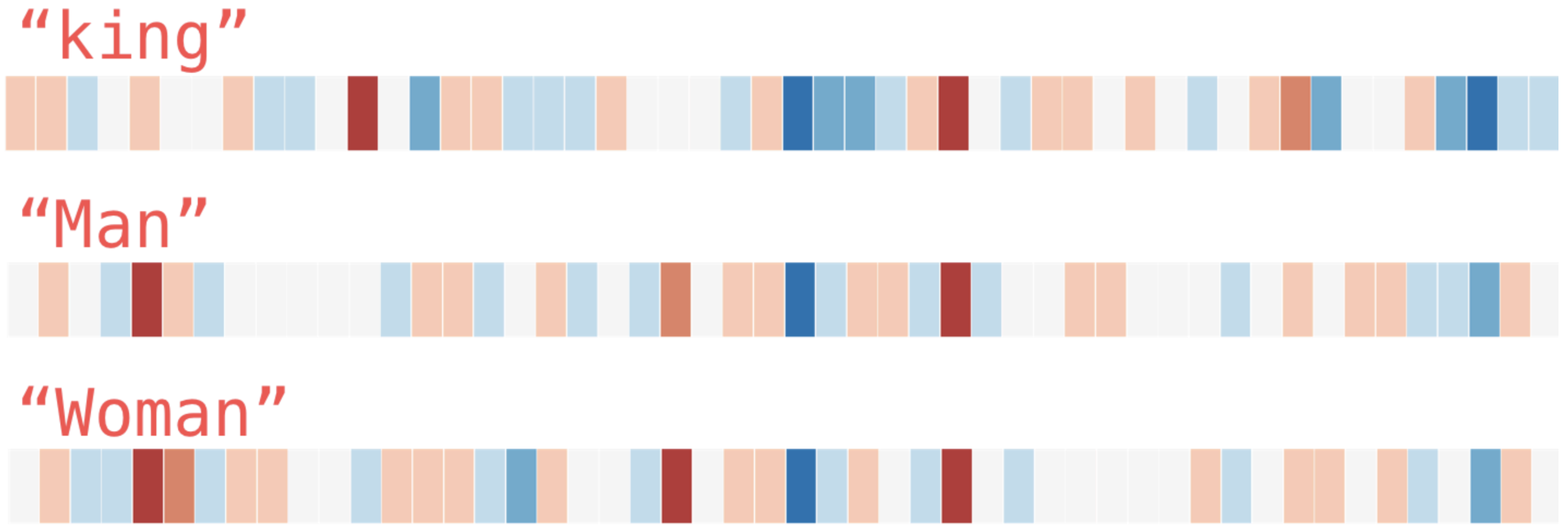
Где находятся векторы слов?



i-й столбец матрицы W' - это вектор i-го слова из словаря!

Почему это работает?

- Обученный word2vec выучил некоторые важные признаки слов (на скрытом слое), и через эти признаки мы получили векторы слов.

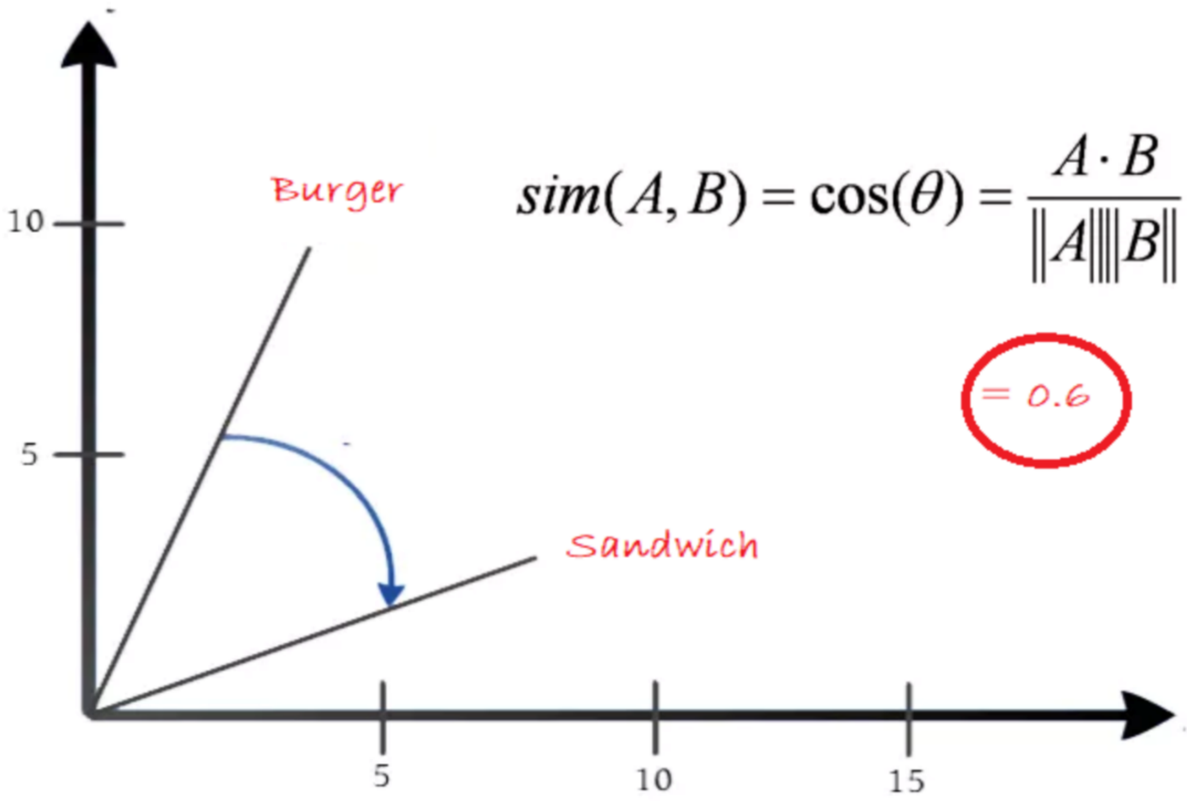


Какие-то компоненты у слов похожи, какие-то нет - это означает, что какие-то смысловые свойства слов совпадают, а какие-то нет.

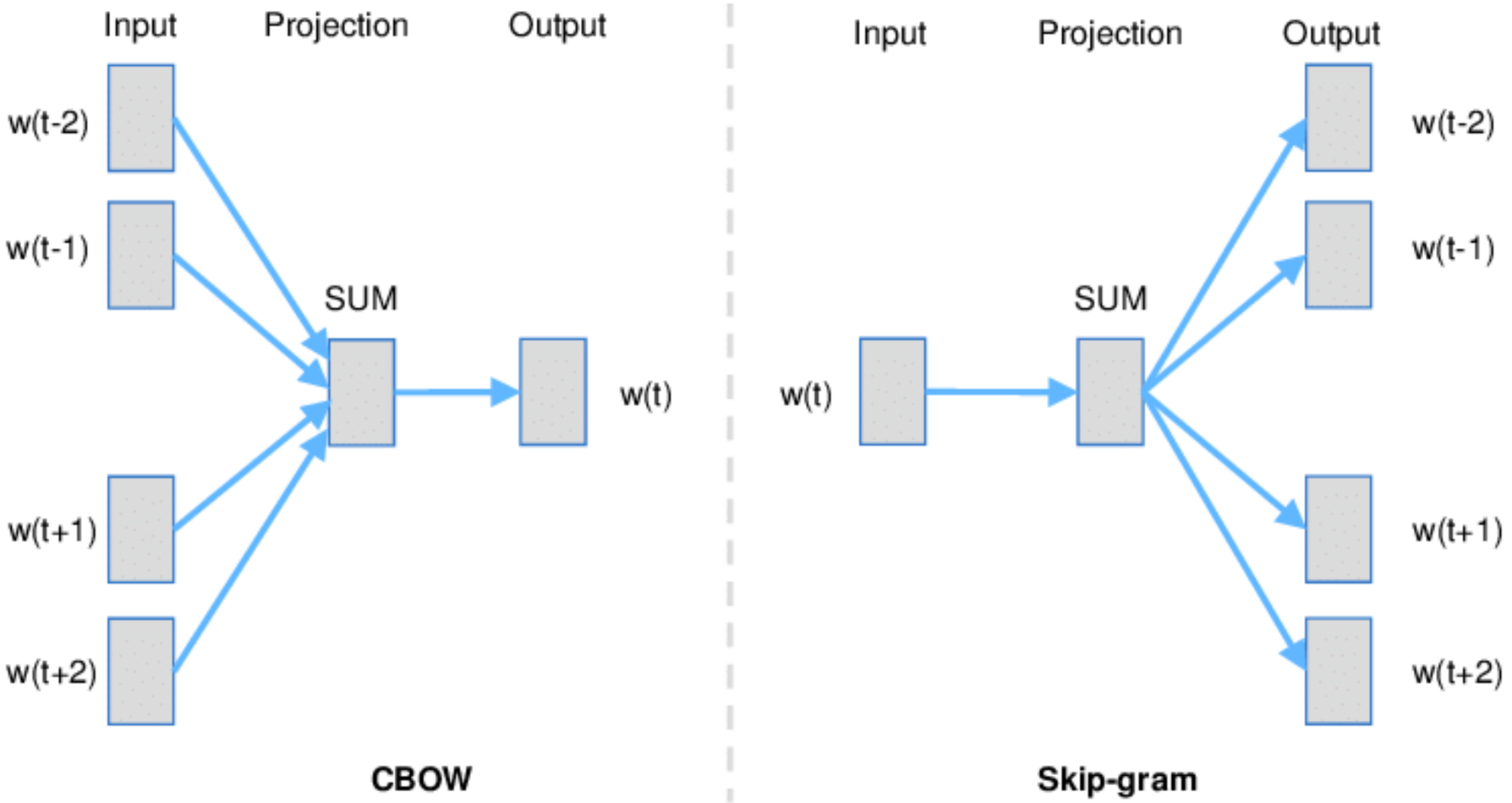
Мера близости слов

В качестве расстояния между словами используется косинусная близость:

$$\rho\left(w_i, w_j\right) = \frac{(w_i, w_j)}{\left|\left|w_i\right|\right| \cdot \left|\left|w_j\right|\right|}$$

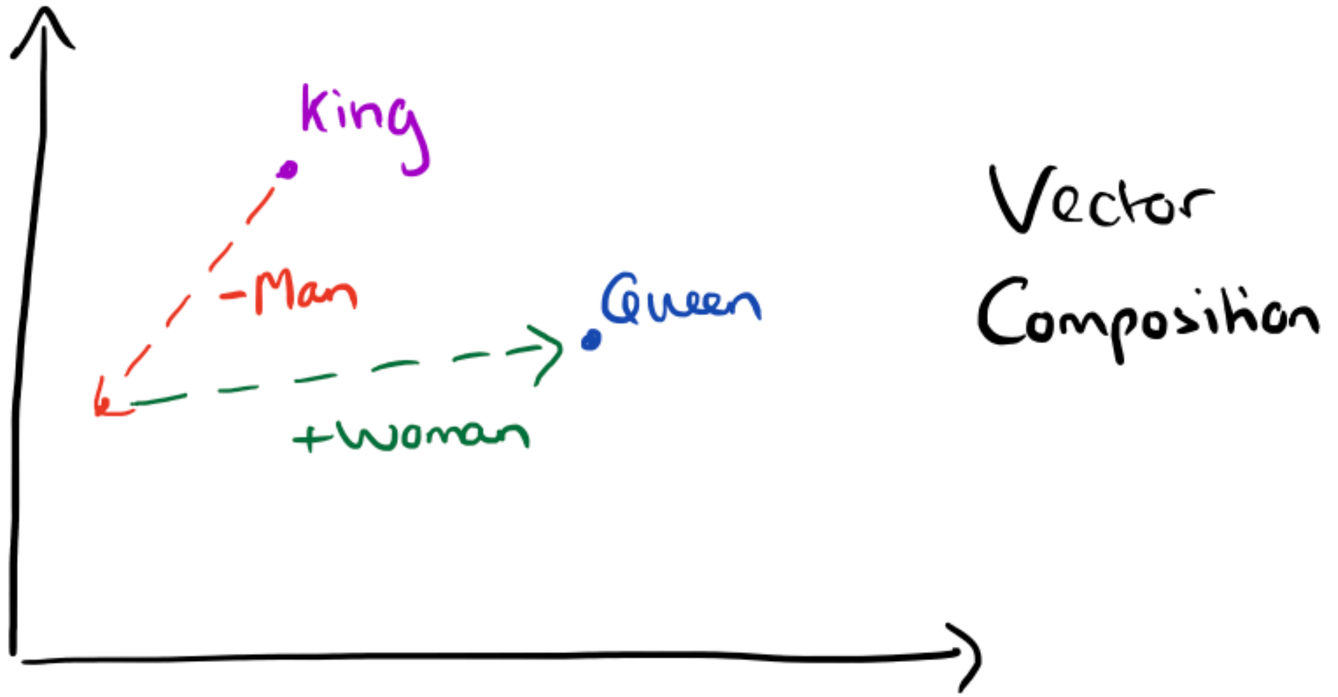


CBOW и SkipGram



Свойства полученных векторов

- С полученными эмбеддингами слов можно проводить математические действия, которые приведут к осмысленным результатам!



Визуализация эмбеддингов