

# СЕМАНТИЧЕСКИЙ АНАЛИЗ: ДИСТРИБУТИВНЫЕ МОДЕЛИ

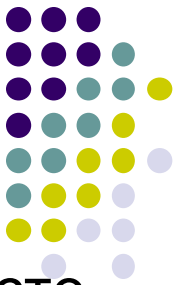
Большакова Елена Игоревна

# СОДЕРЖАНИЕ



1. Дистрибутивная семантика и дистрибутивные модели семантики
2. Счетные (статистические) дистрибутивные модели
  - Дистрибутивные признаки и контексты
  - Семантика векторного пространства
3. Предсказательные (нейросетевые) модели
  - Модель *Word2Vec*: архитектура, обучение
  - Другие модели уровня слов, свойства моделей
  - Обученные модели: ресурс *RusVectors* для РЯ
  - Эволюция нейросетевых моделей
4. Заключение

# ДИСТРИБУТИВНЫЕ МОДЕЛИ СЕМАНТИКИ



- Семантика в языке (как системе) и семантика в тексте
- *Дистрибутивная семантика* – область КЛ, которая занимается вычислением степени семантической близости между единицами текста на основании их распределения в больших массивах текстовых данных.
- *The degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear.*
- По сути, такие модели (частично) восстанавливают семантику единиц по текстам, где они употребляются: (написание слов никак не связано с их смыслом).
- Модели опираются на метод дистрибутивного анализа, применяемый в лингвистике

# ДИСТРИБУТИВНЫЙ АНАЛИЗ



- Дистрибутивный анализ в лингвистике:
  - ❖ метод исследования всех уровней языка:  
фонетика, морфология, синтаксис, семантика
  - ❖ основан на изучении окружения – *дистрибуции, распределения* – единиц в тексте
  - ❖ объединение единиц текста в единицы языка  
(например, морфы-окончания: *землёй – землею*)  
а единиц языка – в классы
- Принцип: разные языковые единицы относятся к одному и тому же классу, если способны замещать друг друга в одних и тех же окружениях (контекстах)
  - Пример: расположение прилагательного после слова *очень* дает класс качественных прилагательных:  
*очень длинный, очень красивый, ..*

# ДИСТРИБУТИВНАЯ ЛЕКСИЧЕСКАЯ СЕМАНТИКА



- **Дистрибутивная гипотеза:** лексические единицы (слова, словосочетания) имеют сходство в значении, если они употребляются в схожих контекстах: *кофе, чай, сок,...*
- Метод дистрибутивного анализа (сходство в значении по контексту, окружению слов) может выявить:
  - Синонимы
  - Антонимы
  - Слова одного семантического классаВсе это – слова с общими *семами* (элементами смысла)
- Модели отличаются способом учета контекста, и в обоих моделях – векторное представление слов:
  - ❖ **Счетные (статистические) модели** на основе векторов (“мешков”) контекстных слов
  - ❖ **Предсказательные модели** на базе нейронных сетей

# ПРИМЕНЕНИЕ ДИСТРИБУТИВНОЙ СЕМАНТИКИ



Модели дистрибутивной лексической семантики для:

- ❖ Построение (и расширение) тезаурусов:  
по коллекции текстов строятся классы близких по семантике слов
- ❖ Кластеризация значений слов и словосочетаний
- ❖ Разрешение неоднозначности слов
- ❖ Поиск близких по значению слов (например, в задаче оценки тональности текста)
- ❖ Расширение запросов в информационном поиске
- ❖ Прикладные задачи КЛ (машинный перевод и др.)
- ❖ и др. – много других задач

# СЧЕТНАЯ МОДЕЛЬ: ПРИМЕР



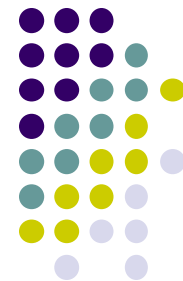
Задача расшифровки иероглифа по его смысловой близости = похожести контекстов

Контекст – строка частот соседних лексем/иероглифов

$$\text{sim}(\text{knife}, \text{cat}) = 0.770$$

(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

# СЧЕТНАЯ МОДЕЛЬ: ПРИМЕР (2)

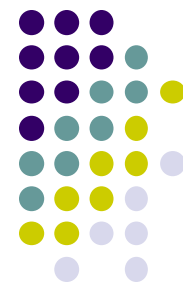


$$\text{sim}(\text{knife}, \text{boat}) = 0.939$$




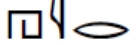

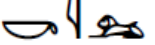





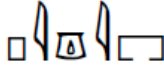

(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0



# СЧЕТНАЯ МОДЕЛЬ: ПРИМЕР (3)



$$\text{sim}(\text{knife}, \text{cat}) = 0.961$$


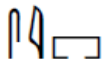

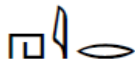

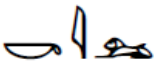





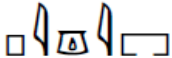

							
(knife)		51	20	84	0	3	0
(cat)		52	58	4	4	6	26
???		115	83	10	42	33	17
(boat)		59	39	23	4	0	0
(cup)		98	14	6	2	1	0
(pig)		12	17	3	2	9	27
(banana)		11	2	2	0	18	0

# СЧЕТНАЯ МОДЕЛЬ: ПРИМЕР (4)



Наиболее близкое слово — *cat*, само слово — *dog*

Оценивалось косинусное расстояние между векторами  
КОНТЕКСТОВ

		get 	see 	use 	hear 	eat 	kill 
knife		51	20	84	0	3	0
cat		52	58	4	4	6	26
<b>dog</b>		<b>115</b>	<b>83</b>	<b>10</b>	<b>42</b>	<b>33</b>	<b>17</b>
boat		59	39	23	4	0	0
cup		98	14	6	2	1	0
pig		12	17	3	2	9	27
banana		11	2	2	0	18	0

# СЧЕТНАЯ (СТАТИСТИЧЕСКАЯ) ДИСТРИБУТИВНАЯ МОДЕЛЬ



Дистрибутивная модель семантики – матрица  $M$  частот совместной встречаемости, каждая её строка представляет собой дистрибуцию (распределение) целевого *терма* относительно контекстов.

$$M = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

*Терм* может пониматься по-разному:  
слово, словосочетание, лемма, морфема, и др.

$(x_{i1}, x_{i2}, \dots, x_{in})$  – вектор распределения  $i$ -го терма, векторное представление

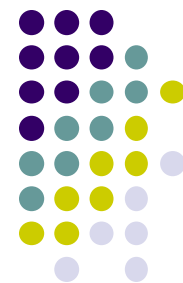
Контекстная матрица  $m \times n$

Строки – вектора термов

Столбцы – *дистрибутивные признаки* (измерения)

Координаты  $x_{i1}, x_{i2}, \dots, x_{in}$  – признаки  $i$ -го терма

# ДИСТРИБУТИВНЫЕ ПРИЗНАКИ



- Признаки основаны на частотах встречаемости
- Часто частоты взвешиваются, а вектора признаков нормализуются, чтобы избежать влияния больших и абсолютных частот

	get	see	use	hear	eat	kill
knife	0.027	-0.024	0.206	-0.022	-0.044	-0.042
cat	0.031	0.143	-0.243	-0.015	-0.009	0.131
dog	-0.026	0.021	-0.212	0.064	0.013	0.014
boat	-0.022	0.009	-0.044	-0.040	-0.074	-0.042
cup	-0.014	-0.173	-0.249	-0.099	-0.119	-0.042
pig	-0.069	0.094	-0.158	0.000	0.094	0.265
banana	0.047	-0.139	-0.104	-0.022	0.267	-0.042

# ГЕОМЕТРИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ МОДЕЛИ



- Строка дистрибутивной матрицы – вектор, описывающий контекст (распределение) слова-терма
- Столбец в таблице – одна из координат пространства векторов
- Важно направление вектора, а не его длина, поэтому обычно применяется нормализация
- Мера близости слов (термов):
  - расстояние между точками на единичной окружности
  - величина угла  $\alpha$  между векторами или косинус угла

$$\cos \alpha = \frac{\sum_{i=1}^n u_i \cdot v_i}{\sqrt{\sum_i u_i^2} \cdot \sqrt{\sum_i v_i^2}}$$

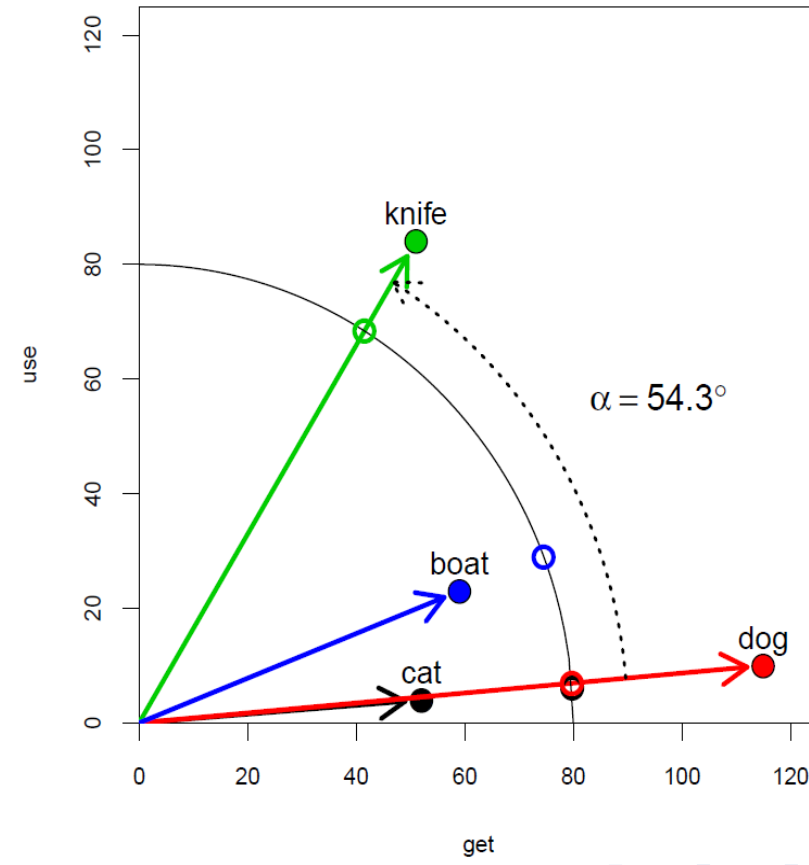


Иллюстрация геометрического семантического расстояния в двух измерениях – “use” и “get”



# КОНТЕКСТ

Контексты различаются по размеру и типу

- Контекст – некоторый фрагмент или единица текста: предложение, абзац, реплика в диалоге, веб-страница, документ.

В последнем случае:

Матрица *Терм-Документ* – встречаемость термов в таком контексте.

- Обычно: *Контекстное окно* фиксированного размера, а признаки целевого терма – соседние термы,

Матрица *Терм-Терм*

	doc <sub>1</sub>	doc <sub>2</sub>	doc <sub>3</sub>	...
boat	1	3	0	...
cat	0	0	2	...
dog	1	0	1	...

	see	use	hear	...
boat	39	23	4	...
cat	58	4	4	...
dog	83	10	42	...

# ТИП КОНТЕКСТА



- Поверхностный контекст – окно из слов (символов) вокруг целевого термина; его параметры:
  - размер  $k$  окна (в словах или в символах)
  - симметричное или одностороннее окно
  - равномерное или «треугольное» (т.е. взвешивание, основанное на линейном расстоянии между целевым и контекстным термом)
  - (дополнительно): ограничено ли окно предложениями или другими текстовыми единицами
- Более глубокий контекст, часто – синтаксический, когда контекстный терм связан с целевым термом синтаксической зависимостью; возможные параметры:
  - тип синтаксической зависимости
  - максимальная длина пути зависимости

# ПОСТРОЕНИЕ КОНТЕКСТНОЙ МАТРИЦЫ



1. Лингвистическая предобработка текста, выделение контекстных признаков (слов или др.), при этом по минимуму – токенизация текста, но еще может быть:
  - ✓ частеречная разметка / лемматизация / стемминг
  - ✓ поверхностный синтаксический анализ
2. Вычисление частот признаков
3. Взвешивание признаков – чтобы снизить значимость абс.частот и уменьшить вклад менее значимых признаков
  - ✓ Логарифмическое взвешивание:  $x' = \log(x+1)$
  - ✓ Мера *TF-IDF* – повышает значимость редких событий
  - ✓ Статистич. меры связности (ассоциации): *t-score*, *MI*, *PPMI* – *положительная поточечная MI* (хорошо работает):  
 $PPMI=MI, MI \geq 0$  и  $PPMI=0, MI < 0$
4. Нормализация векторов слов



# СЧЕТНЫЕ ДИСТРИБУТИВНЫЕ МОДЕЛИ: ПРИМЕРЫ

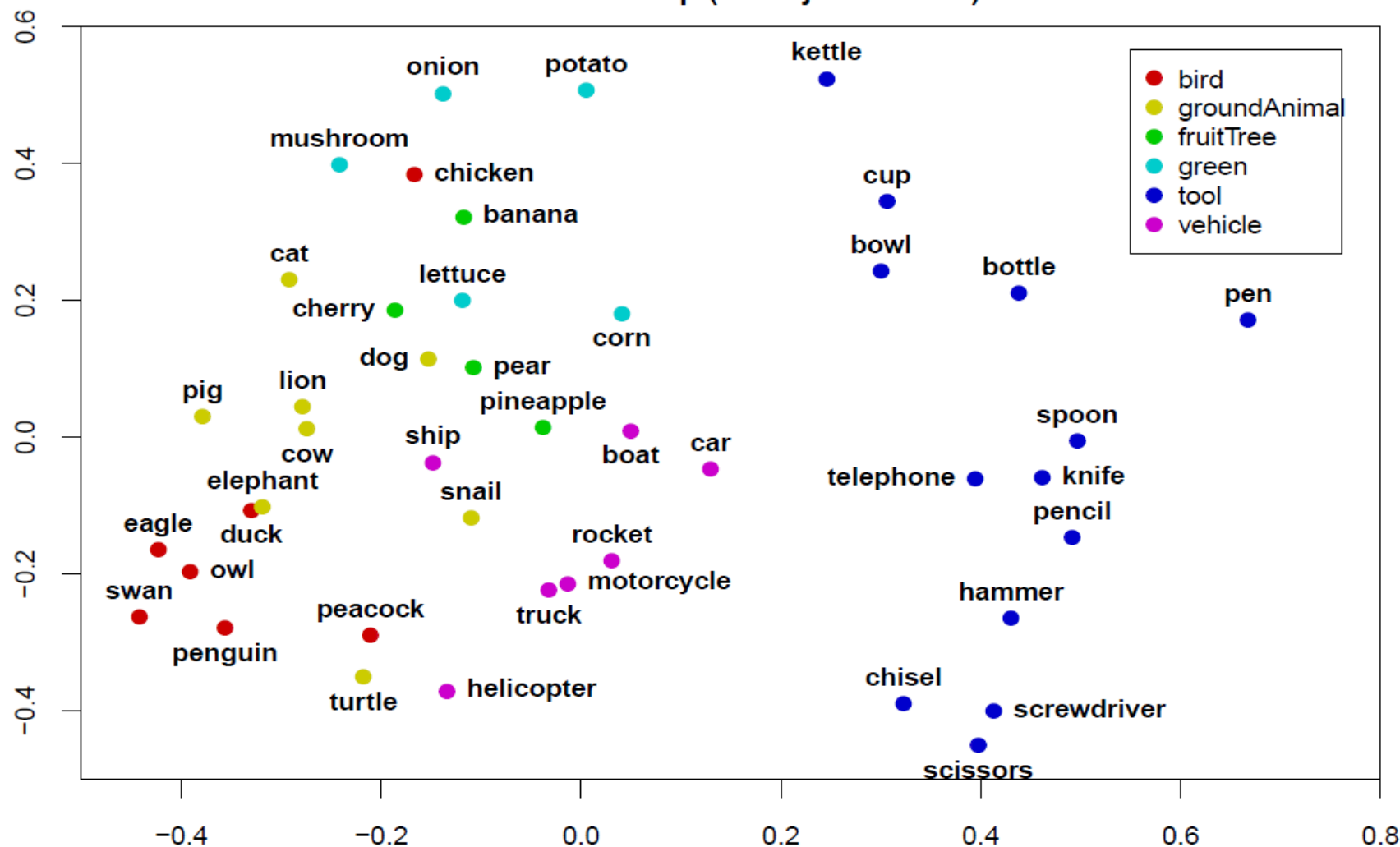


- Данные по корпусу British National Corpus:  
чем больше расстояние – тем дальше (по смыслу) сосед
- Ближайшие соседи слова *dog*
  - *girl* (45.5), *boy* (46.7), *horse* (47.0), *wife* (48.8), *baby* (51.9), *daughter* (53.1), *side* (54.9), *mother* (55.6), *boat* (55.7), *rest* (56.3), *night* (56.7), *cat* (56.8), *son* (57.0), *man* (58.2), *place* (58.4), *husband* (58.5), *thing* (58.8), *friend* (59.6)
- Ближайшие соседи слова *school*
  - *country* (49.3), *church* (52.1), *hospital* (53.1), *house* (54.4), *hotel* (55.1), *industry* (57.0), *company* (57.0), *home* (57.7), *family* (58.4), *university* (59.0), *party* (59.4), *group* (59.5), *building* (59.8), *market* (60.3), *bank* (60.4), *business* (60.9), *area* (61.4), *department* (61.6), *club* (62.7), *town* (63.3), *library* (63.3), *room* (63.6), *service* (64.4), *police* (64.7)

# ДИСТРИБУТИВНЫЕ МОДЕЛИ: СЕМАНТИЧЕСКИЕ КАРТЫ



Semantic map (V-Obj from BNC)

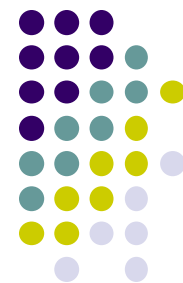


# СЕМАНТИЧЕСКИЕ СХОДСТВО И АССОЦИАЦИЯ



- Слово *a* имеет большее семантическое (смысловое) сходство/близость с/к *b*, чем с/к *c*, если в дистрибутивном векторном пространстве вектор  $\uparrow a$  ближе к  $\uparrow b$ , чем к  $\uparrow c$
- Семантическим сходством обладают два слова, имеющие большое число заметных общих черт:
  - Синонимия: *машина/автомобиль*, квази: *лампа/ночник*
  - Гиперонимия: *автомобиль/транспорт*
  - Ко-гипонимия : *грузовик/фургон/ легковушка*
- Семантическая ассоциация: два семантически близких слова, но их сходство не является обязательным условием:
  - Функция: *машина/ездить*
  - Меронимия: *машина/колесо*
  - Местоположение: *машина/гараж*
  - Атрибут : *машина/быстрая*
  - Неявные ассоциации: *цветок /красота*

# ПРОБЛЕМЫ СЧЕТНЫХ ДИСТРИБУТИВНЫХ МОДЕЛЕЙ



- Контекстная матрица  $M$  зачастую получается очень большой и крайне разреженной
  - *Google Web1T5*: матрица размера 1 млн.  $\times$  1 млн. : триллион клеток и ненулевые значения – только 0,05 %
- Как понизить размерность матрицы (векторного пространства) и сделать вектора менее разреженными?
  - Латентный семантический анализ, сингулярное разложение матриц *SVD* (*Singular Values Decomposition*) и метод *PCA* (метод главных компонент)  
Но: Вычислительная сложность
- Другой способ построения векторного пространства слов меньшей размерности – предсказательные дистриб. модели
- Оба вида дистрибут. моделей строят *word embeddings* (вложение, погружение слова в пр-во числовых векторов)

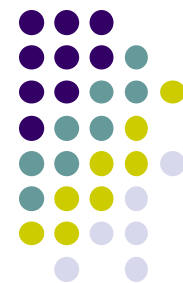
# ПРЕДСКАЗАТЕЛЬНЫЕ ДИСТРИБУТИВНЫЕ МОДЕЛИ



- Комбинирование дистрибутивной векторной семантики с вероятностными языковыми моделями
- Векторное пространство строится при обучении (чаще – нейронной сети) для решения задачи языкового моделирования, т.е. предсказания соседних слов
- Каждому уникальному слову соответствует свой вектор – *word embedding, distributive word representation* (*эмбеddинг*, векторное представление слова),
- Поиск оптимального векторного представления слов – при максимизации следующей функции:

$$J(\theta) = \frac{1}{T} \sum_{t=1}^T \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t)$$

# ПРЕДСКАЗАТЕЛЬНЫЕ МОДЕЛИ: ОСОБЕННОСТИ



- Обучение (*self-supervision*) на базе очень большой неразмеченной текстовой коллекции
- В итоге каждое слово представляется неразрезанным вектором низкой размерности (обычно 300-500 элементов)
- Как и в счетных дистрибутивных моделях:
  - в построенном векторном пространстве смысл имеют только расстояния между векторами, а не сами вектора
  - строится семантическое пространство слов
- Наиболее известны предсказательные модели *Word2Vec, Glove, FastText* (модели на уровне слов)
- Одна из первых моделей – *Word2Vec* (Google, 2013)
  - нейронная двухслойная сеть прямого распространения
  - предсказание как классификация, на основе статистики совместной встречаемости слов, с учетом и без учета близости слов в контексте

# Word2Vec : ВАРИАНТЫ



Две нейросетевые архитектуры для однослойного персептрона с логистической функцией активации

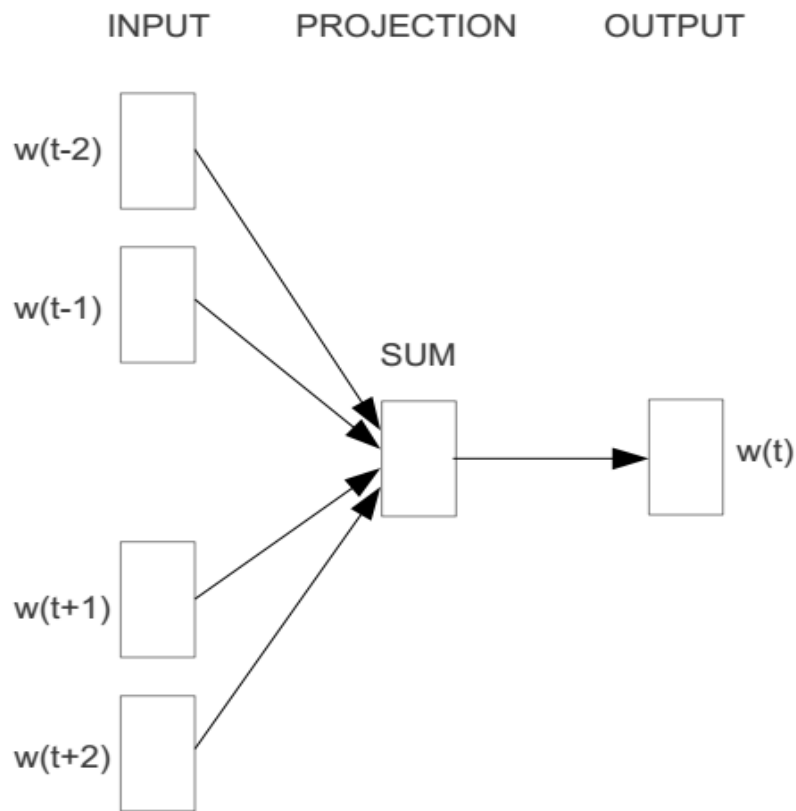
Параметр – контекстное окно (3-10 слов, по умолчанию 5)

- **CBOW** (*Continuous Bag-of-Words*)
  - Предсказание слов при заданном окне контекста (прямая задача языкового моделирования)
  - Контекст – «непрерывный мешок слов», например, 4 ближайших слова: 2 предыдущих, 2 последующих
  - Работает быстрее + лучше для больших корпусов (миллионы и миллиарды слов), т.к. частота слов выше
- **SkipGrams** (*Continuous Skip-n-Grams*)
  - Предсказание близкого контекста при заданном слове
  - Контекст – *n-граммы*, с учетом пропусков слов
  - Медленнее, но лучше работает для редких слов и не слишком больших корпусов (<100 млн.токенов)

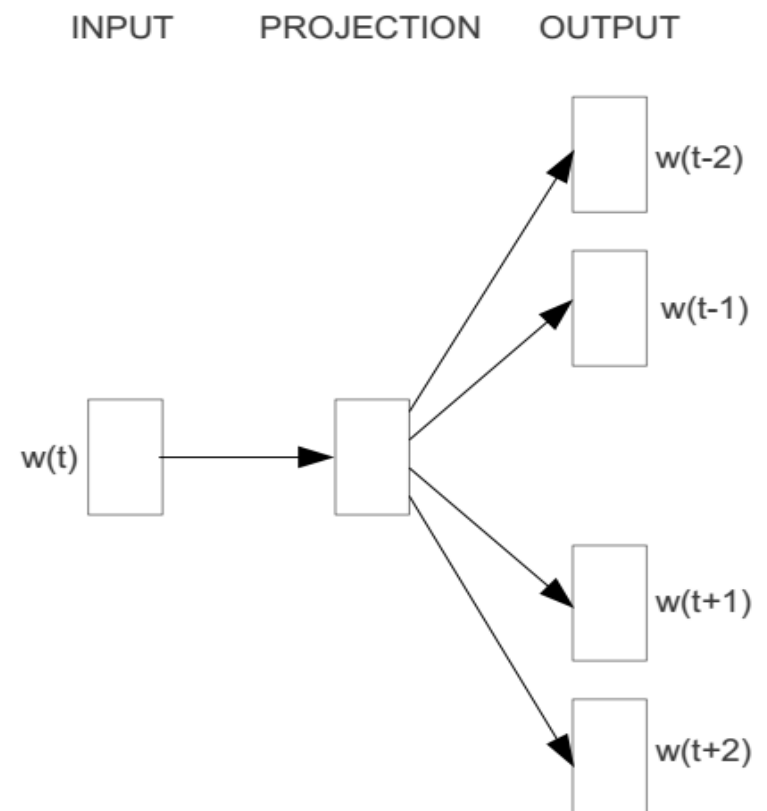
# Word2Vec: АРХИТЕКТУРЫ ОБУЧЕНИЯ



Общий вид при размере окна = 2:



**CBOW**



**Skip-gram**



# Word2Vec: СЛОИ СЕТИ



- Вход: *one-hot vector* (*one-hot encoding*) : для словаря размера  $T$  слово кодируется булевым вектором ровно с одной единичной компонентой в позиции, равной номеру слова в словаре
  - $dog = (0, 0, 0, 0, 1, 0, 0, 0, 0, \dots)^T$
  - $cat = (0, 0, 0, 0, 0, 0, 0, 1, 0, \dots)^T$
  - $eat = (0, 1, 0, 0, 0, 0, 0, 0, 0, \dots)^T$
- Выходной вектор: каждая координата – вероятность того, данное слово будет следующим (в тексте), т.е. предсказание
- Скрытый слой (*Projection layer*) выделяет вектор слова: выученные веса представляют вектора слов
- Вероятности на выходе, применяется *softmax*: повышает максимальную величину и «прижимает» меньшие величины

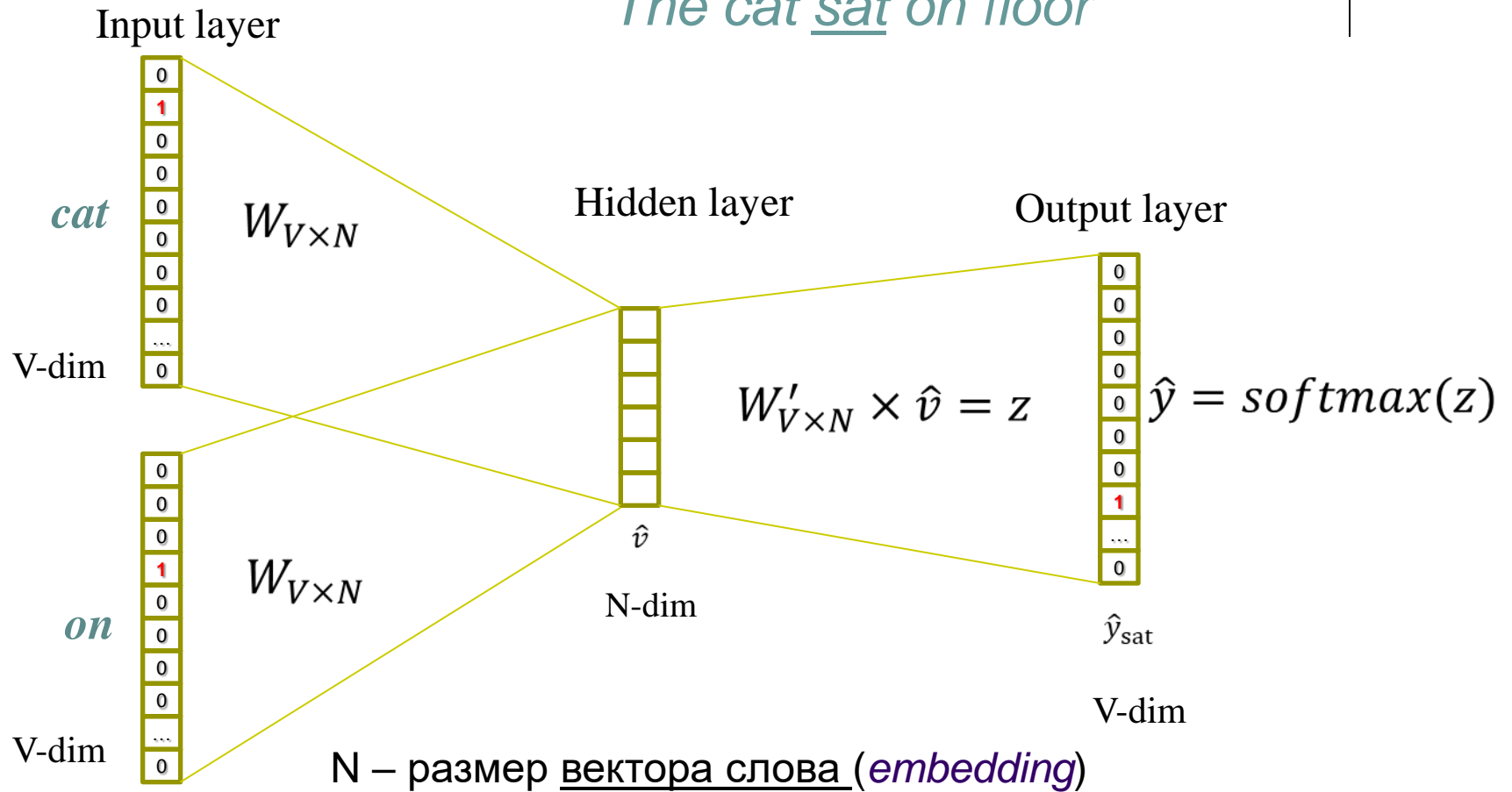
$$p_i = \text{softmax}(s_i, \vec{s}) = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

[1,2,3,4,1,2] -> softmax: [0.024, 0.064, 0.175, **0.475**, 0.024, 0.064]

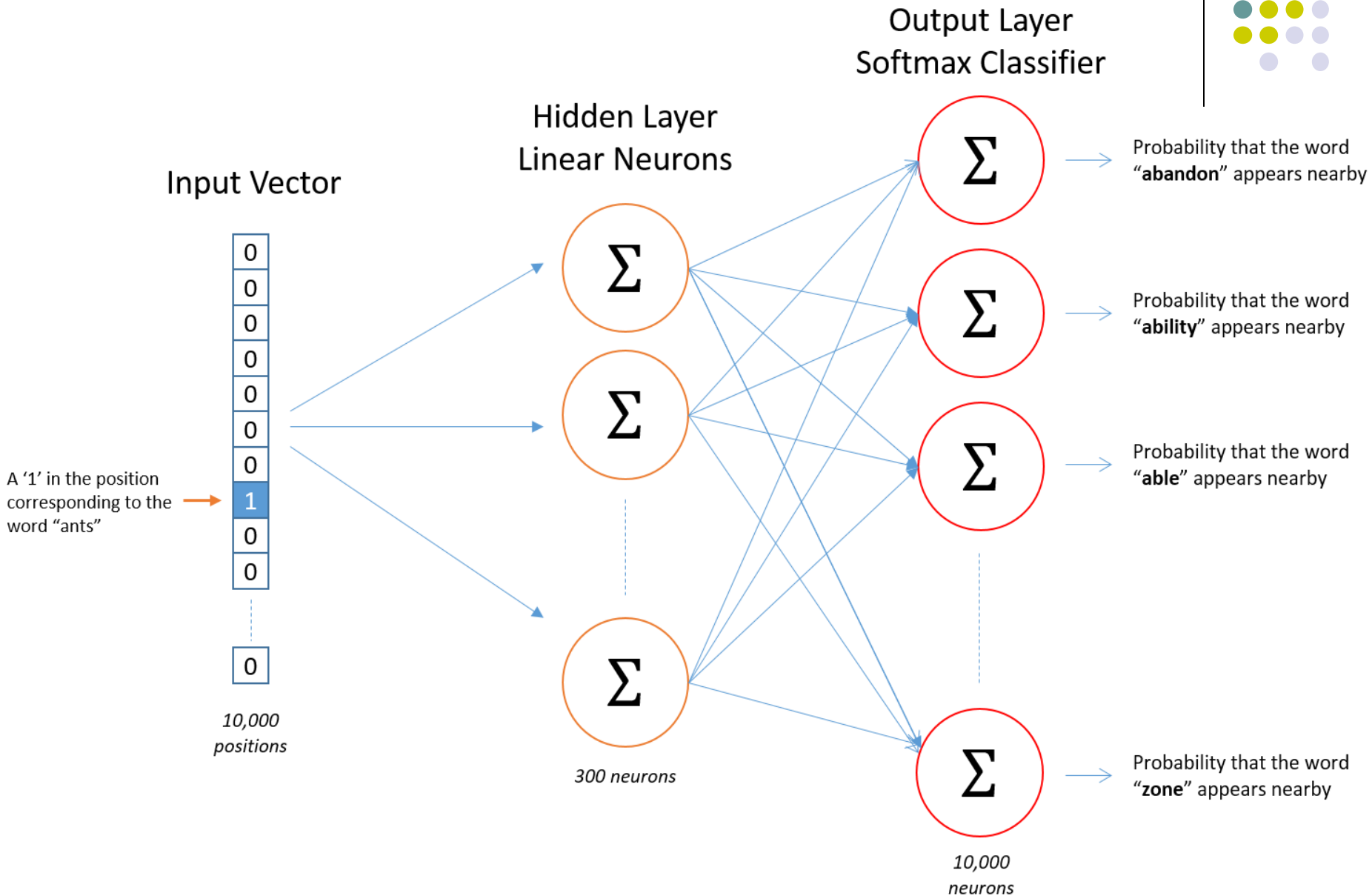


# Word2Vec: CBOW

*The cat sat on floor*



# Word2Vec: Skip-gram



# ОСОБЕННОСТИ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ *Word2Vec*



Матрицы большого размера и долгая обработка, поэтому:

- *Subsampling frequent words* (подвыборка, субдискретизация): т.к. частотные слова не столь важны, они выкидываются из текста с вероятностью, пропорциональной их частоте, что ускоряет обучение и улучшает качество модели.
- *Negative sampling* (негативное сэмплирование): случайная выборка и обновление отрицат. примеров
- *Hierarchical Softmax* Иерархический софтмакс хорошо подходит для создания лучшей модели относительно редких слов, негативное сэмплирование лучше моделирует более частотные слова.
- С матем. точки зрения: сингулярное разложение матрицы *PPMI* (поточечных взаимных информаций) слов
- Эмбединги – побочный эффект обучения языковой модели

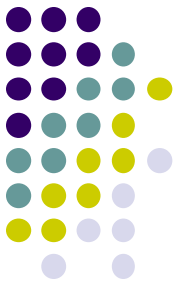
# ПАРАМЕТРЫ ОБУЧЕНИЯ *Word2Vec*



- ❖ Гиперпараметры обучения: *CBOW* или *Skip-gram*
  - Размер окна
  - Размер векторов (итогового векторного пространства)
  - *Negative sampling* или же *Hierarchical Softmax*
- ❖ Эвристическая информация из опыта обучения:
  - Не существует однозначно лучшего способа обучения, т.е. наилучшей комбинации
    - архитектура + конкретные гиперпараметры
  - Гиперпараметры часто оказываются более важными для получения лучшей модели, чем тип архитектуры и увеличение объема данных для обучения
  - Размер окна: для *Skip-gram* оптимальный размер около 10, для *CBOW* – в районе 5
- Недостаток модели *Word2Vec* – нет векторов для редких слов и слов, не встречающихся в обучающей коллекции

# Word2Vec : ПРИМЕРЫ

Модель, обученная по интернет-текстам  
Вывод по мере близости (контекстов)



Enter word: **avito**

Word Cosine distance:

— *awito* 0.693721

*avumo* 0.675299

*fvito* 0.661414

*avuma* 0.659454

*irr* 0.642429

*овumo* 0.606189

*avimo* 0.598056

Enter word: **mail**

— *rambler* 0.777771

*meil* 0.765292

*inbox* 0.745602

*maill* 0.741604

*yandex* 0.696301

*maili* 0.675455

*myrambler* 0.674704

*zmail* 0.657099

*mefr* 0.655842

*jandex* 0.655119

*gmail* 0.652458

*вкmail* 0.639919

# НЕЙРОННАЯ МОДЕЛЬ *FastText*



- *FastText* (Facebook, 2014) создавалась для задач классификации текстов + исправление недостатка *Word2Vec*
- Модель на базе *subword embeddings*: строит векторные представления N-грамм символов (триграмм)
- Эмбеддинг слова – усредненная сумма векторов всех его N-грамм, и можно построить вектор редкого/отсутствующего слова (N-граммы встречаются чаще, чем целиком слова)
- Параметры: – *CBOW* или *Skip-gram*, – величина N  
– размер окна/контекста и размер итоговых векторов  
– наименьшее допустимое количество символов в слове
- *FastText* медленнее обучается, чем *Word2Vec* (чем сложнее входные данные, тем больше время обучения)
- Эта модель лучше для языков с богатой морфологией (в том числе РЯ), когда в текстах нет многих словоформ

# ДИСТРИБУТИВНАЯ МОДЕЛЬ *Glove*



*Glove (Global Vectors)* – еще одна дистрибутивная модель семантики уровня слов

- Изначальная цель – построение именно эмбедингов
- Получена обучением, направленным на геометрические (линейные) соотношения между векторами
- Не использует нейронные сети, учитывает совместную встречаемость слов (статистику)

Общая проблема всех дистрибутивных моделей уровня слов:

лексич. неоднозначность (омонимия, полисемия):

- *банка* : банк; госбанк; сбербанк; банкир; заемщик;
- *белка* : белок; фермент; молекула; полисахарид;
- *парка* : парк; сквер; парковый; рекреация; фонтан;
- *сталь* : стать; становиться; являться; послужить;
- *пила* : пить; запить; выпить; потягивать; наливать;



# ГЕОМЕТРИЧЕСКИЕ СВОЙСТВА ВЕКТОРНОГО ПРОСТРАНСТВА

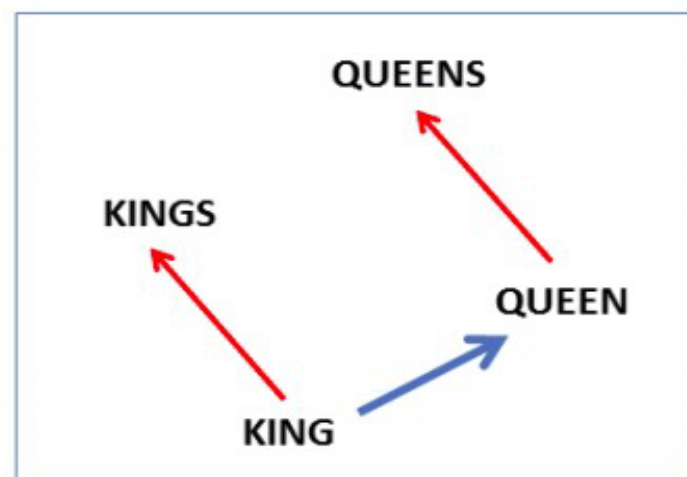
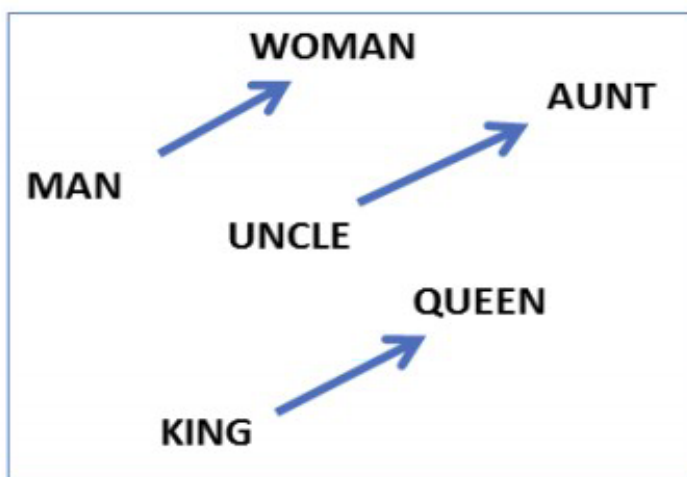


Аналогии, выявляемые путем операций  
с векторами слов:  $king + (woman - man) = queen$

$kings - king = queens - queen$

Вычисленный вектор не точно равен *queen*, но это  
наиболее близкий результат

- ❖ Важно: Геометрическим соотношениям соответствуют  
семантические соотношения между словами



# ВАЛИДАЦИЯ ОБУЧЕННОЙ МОДЕЛИ



- Проводится после обучения на коллекции
- Тесты – вопросы вида:  
*Что относится к С так же, как В относится к А?*  
Вычисляется вектор  $X = \text{vector}(B) - \text{vector}(A) + \text{vector}(C)$   
Ответ – вектор, ближайший по косинусной мере к  $X$
- Два типа вопросов:
  - Семантические: *Что относится к Германии так же, как Париж относится к Франции? (Берлин)*
  - Грамматические: *Что относится к small так же, как biggest относится к big? (smallest)*
- Тестовые данные для английского: четверки слов (3 известных + 1 предсказываемое), состояются из пар слов
  - 8869 семантических вопросов
  - 10675 грамматических вопросов

# ПРИМЕРЫ ПАР СЛОВ ИЗ ТЕСТОВ



## Семантические отношения

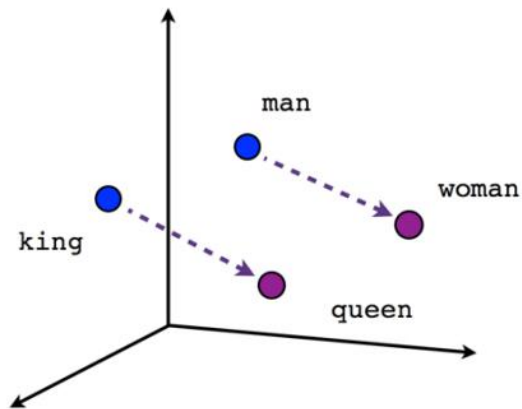
	Пара 1		Пара 2	
Известные столицы	Athens	Greece	Oslo	Norway
Столицы	Astana	Kazakhstan	Harare	Zimbabwe
Валюта	Angola	kwanza	Iran	rial
Город в штате	Chicaga	Illinois	Stockton	California

## Грамматические отношения

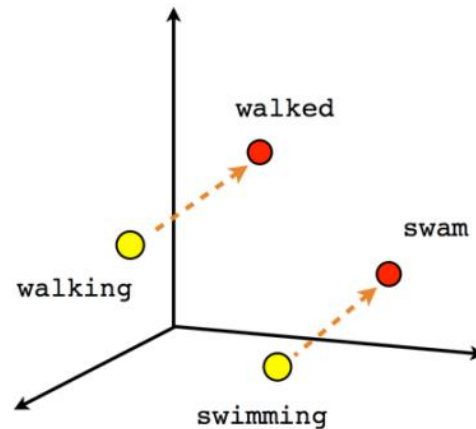
	Пара 1		Пара 2	
Прил.-Нар.	apparent	apparently	rapid	rapidly
Отрицание	possibly	impossibly	ethical	Unethical
Мн. число	mouse	mice	dollar	dollars
Время глагола	walking	walked	swimming	swam
Сравнение	great	greater	tough	tougher



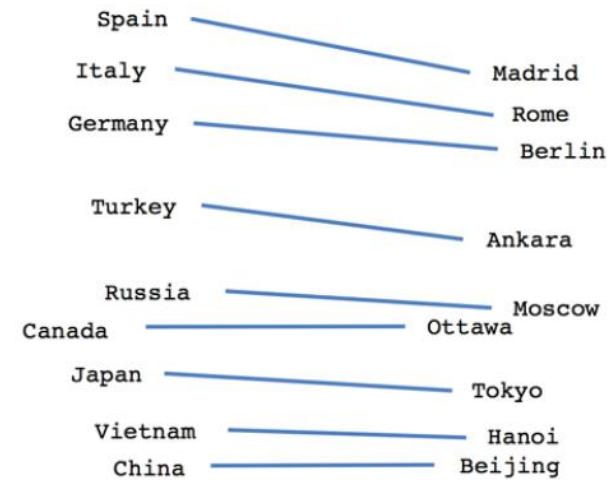
# ГЕОМЕТРИЧЕСКИЕ СООТНОШЕНИЯ ПАР СЛОВ



Male-Female



Verb tense



Country-Capital

# РЕАЛИЗАЦИИ НЕЙРОННЫХ ДИСТРИБУТИВНЫХ МОДЕЛЕЙ



- ❖ Модель *Word2Vec*
  - Исходный код: <https://github.com/tmikolov/word2vec>
  - *Gensim*  
<https://radimrehurek.com/gensim/models/word2vec.html>
  - В пакетах нейронных сетей *Torch*, *TensorFlow*, *Theano*
- ❖ Предобученные модели для **РЯ**: [Rusvectors.ru](http://Rusvectors.ru)
  - *Word2Vec*, *FastText*
  - На разных больших текстовых массивах:  
Википедия, *НКРЯ*, корпус *Тайга* и др.
  - Обычно предварительная лемматизация для  
уменьшения размерности данных
  - Веб-интерфейс: возможность работы на сайте

# *RusVectors:* СХОДСТВО ВЕКТОРОВ СЛОВ



## Semantic associates for *стол* (ALL)

### Ruscorpora and Russian Wikipedia

1. столик 0.679
2. табурет 0.526
3. табуретка  
0.515
4. подоконник  
0.501
5. диван 0.491
6. стул 0.484
7. кровать 0.476
8. тумбочка 0.447
9. парта 0.439
10. кушетка 0.428

### Ruscorpora

1. столик 0.794
2. подоконник  
0.642
3. табуретка  
0.637
4. табурет 0.623
5. диван 0.582
6. кровать 0.573
7. стул 0.570
8. кушетка 0.561
9. тумбочка 0.561
10. кресло 0.552

### Web corpus

1. столик 0.637
2. стул 0.570
3. табурет 0.554
4. поднос 0.525
5. тумбочка 0.517
6. табуретка  
0.497
7. обеденный  
0.490
8. кушетка 0.482
9. кресло 0.479
10. сервировочный  
0.470

# Rus Vectors: СМЫСЛОВЫЕ СИНОНИМЫ



рисунок\_NOUN

## НКРЯ

1. изображение<sub>NOUN</sub> 0.68
2. акварель<sub>NOUN</sub> 0.66
3. гравюра<sub>NOUN</sub> 0.65
4. картинка<sub>NOUN</sub> 0.65
5. орнамент<sub>NOUN</sub> 0.64
6. акварельный<sub>ADJ</sub> 0.64
7. эскиз<sub>NOUN</sub> 0.63
8. композиция<sub>NOUN</sub> 0.62
9. виньетка<sub>NOUN</sub> 0.62
10. миниатюра<sub>NOUN</sub> 0.61

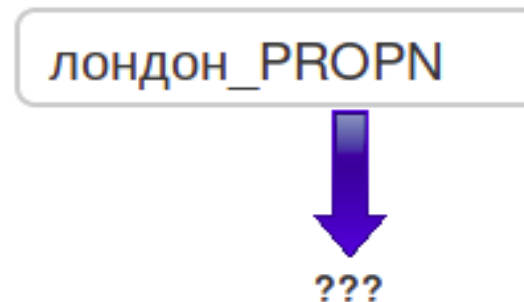
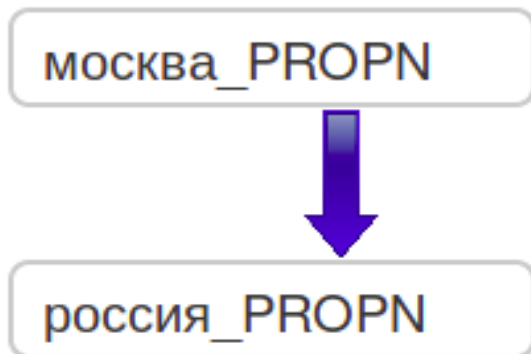


## НКРЯ и Wikipedia

1. гравюра<sub>NOUN</sub> 0.69
2. иллюстрация<sub>NOUN</sub> 0.67
3. эскиз<sub>NOUN</sub> 0.64
4. акварель<sub>NOUN</sub> 0.63
5. фотография<sub>NOUN</sub> 0.63
6. картинка<sub>NOUN</sub> 0.63
7. узор<sub>NOUN</sub> 0.61
8. акварельный<sub>ADJ</sub> 0.60
9. изображение<sub>NOUN</sub> 0.59
10. рисовать<sub>VERB</sub> 0.58



# RusVectores: АНАЛОГИИ, ПРИМЕР - 1



## НКРЯ

1. **англия**<sub>PROP</sub> 0.72
2. **франция**<sub>PROP</sub> 0.67
3. **европа**<sub>PROP</sub> 0.67
4. **британия**<sub>PROP</sub> 0.63
5. **британский**<sub>ADJ</sub> 0.61



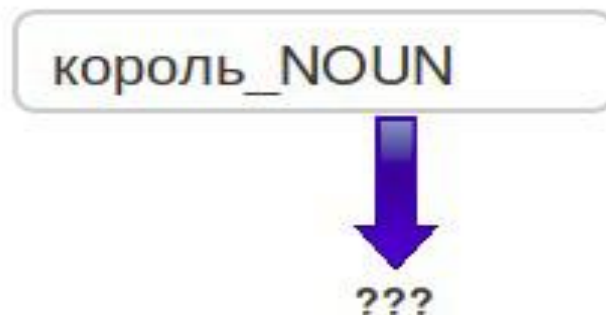
## НКРЯ и Wikipedia

1. **англия**<sub>PROP</sub> 0.58
2. **европа**<sub>PROP</sub> 0.54
3. **великобритания**<sub>PROP</sub> 0.52
4. **страна**<sub>NOUN</sub> 0.48
5. **франция**<sub>PROP</sub> 0.47





# RusVectores: АНАЛОГИИ, ПРИМЕР - 2



## НКРЯ

1. герцог<sub>NOUN</sub> 0.69
2. королева<sub>NOUN</sub> 0.65
3. дофин<sub>NOUN</sub> 0.61
4. принц<sub>NOUN</sub> 0.61
5. королевство<sub>NOUN</sub> 0.60



## НКРЯ и Wikipedia

1. королева<sub>NOUN</sub> 0.75
2. королю<sub>NOUN</sub> 0.58
3. императрица<sub>NOUN</sub> 0.58
4. принцесса<sub>NOUN</sub> 0.57
5. государь<sub>NOUN</sub> 0.56

# *RusVectors:* ВЫЧИТАНИЕ ВЕКТОРОВ ПРИМЕР




+

стул\_NOUN

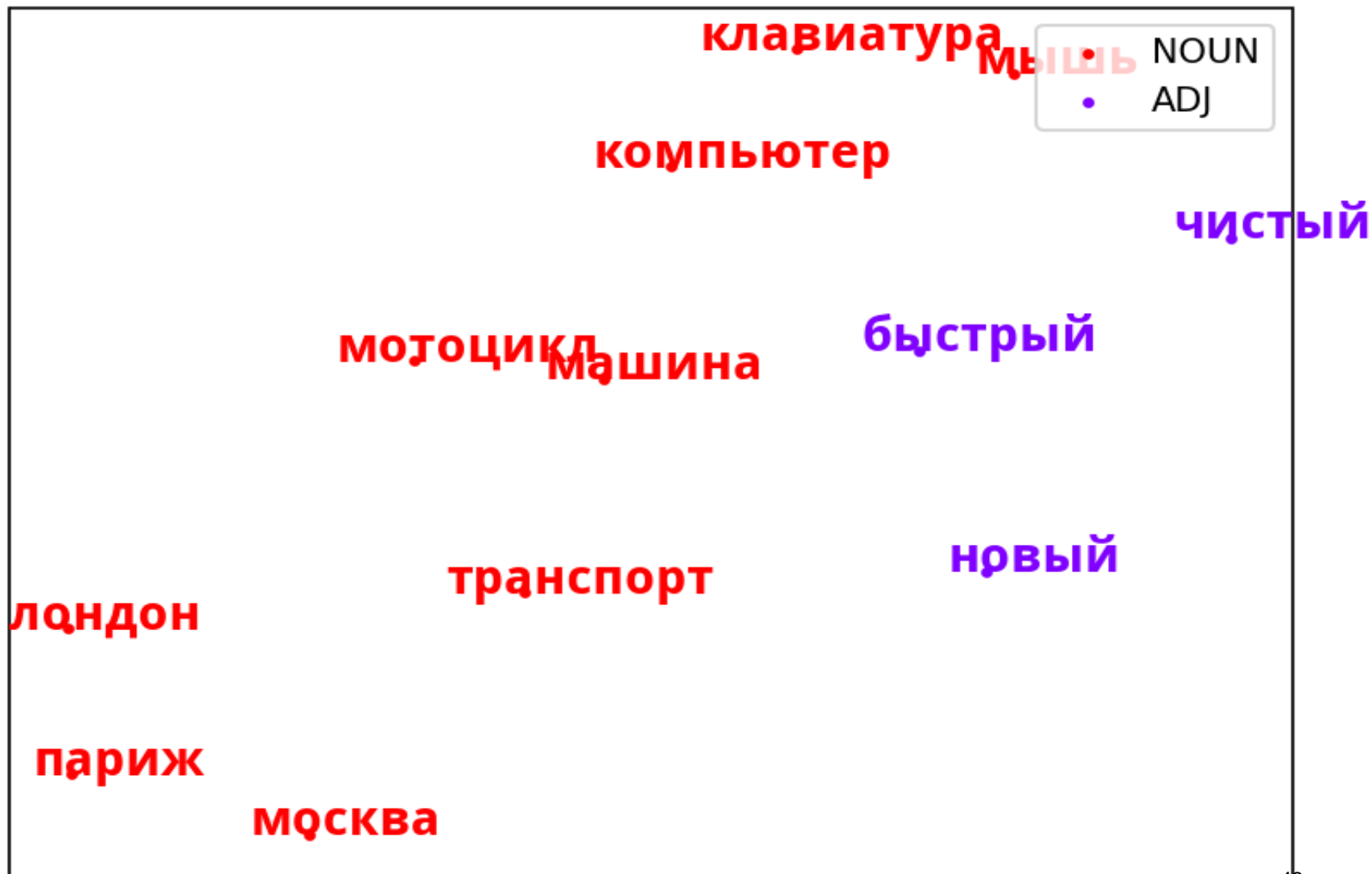
-

спинка\_NOUN

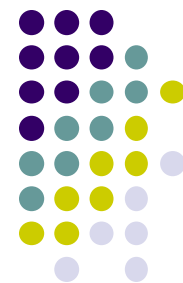
## НКРЯ

1. табуретка<sub>NOUN</sub> 0.37
2. табурет<sub>NOUN</sub> 0.35 
3. стултраница<sub>NOUN</sub> 0.35
4. корточки<sub>NOUN</sub> 0.34
5. клавикорда<sub>NOUN</sub> 0.33

# RusVectors: ВИЗУАЛИЗАЦИЯ

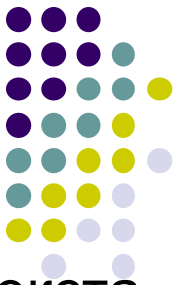


# ЭВОЛЮЦИЯ НЕЙРОСЕТЕВЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ



- Модели уровня слов (*Word2Vec* и др.) дают *статические эмбе́ддинги*, т.е. один фиксированный вектор для каждого слова входного словаря, что не вполне учитывает контекст и многозначность слов
- Дальнейшее развитие дистрибут. моделей семантики:
  - более глубокий учет контекста
  - учет уровня символов (частично: в *FastText*)
  - более сложные нейросетевые архитектуры и обучение
- Разработан ряд моделей, вырабатывающих *контекстуализированные эмбе́ддинги* (*contextualized word embeddings*), например: для слова *среда* из сочетаний *среда обитания* и *в ближайшую среду* получаются разные эмбе́ддинги
- Наиболее известны *ELMo* (2018), *BERT* (2018), *GPT* (2020)

# МОДЕЛЬ *ELMo*

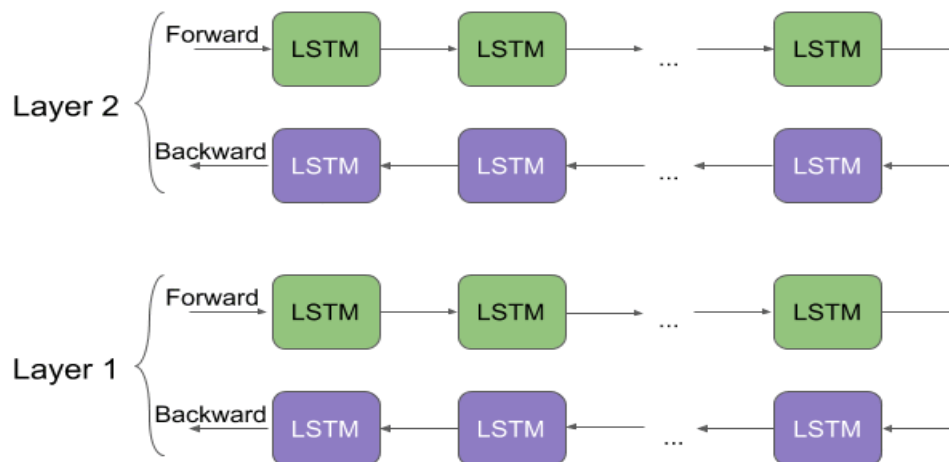


- Основной недостаток нейронных сетей прямого распространения (*FFNN*) – фиксированная длина контекста
- *Рекуррентные* нейронные сети (*RNN*, 2010) могут работать с длинными последовательностями  
(их скрытое состояние в принципе может представлять информацию обо всех предыдущих словах текста),  
т.е. практически полноценное языковое моделирование
- *ELMo* (*Embeddings from Language Models*) – модель на базе рекуррентной сети *Bi-LSTM* (двунаправленная модель долгой краткосрочной памяти)
  - Основа: две взаимосвязанные *Bi-LSTM*:  
*forward language model* и *backward language model*
  - Слова текста обрабатываются как последовательности символов, и возможен учет внутренней структуры слова (улавливаются связи вида *простое* и *простота* )

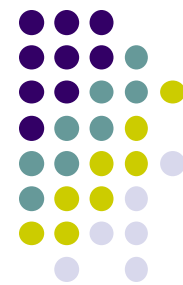
# МОДЕЛЬ *ELMo* : АРХИТЕКТУРА



- Два слоя, работа первого слоя:
  - Вход для прямого прохода: слово и контекст перед ним
  - Вход для обратного прохода: слово и контекст после
  - Результат проходов – промежуточные векторы слов
- Второй слой работает аналогично первому, на входе – промежуточные векторы
- Результирующее представление – взвешенная сумма векторов исходных слов и двух промежуточных векторов слов



# НЕЙРОСЕТЕВАЯ МОДЕЛЬ *BERT*



- Рекуррентные нейронные сети высокочувствительны по памяти и времени обучения, многократная передача информации может приводить к ее потере
- Нейросетевая архитектура *Transformer* (2017) – работа с последовательностями только за счет *механизма внимания*, без рекуррентности
- *BERT* (*Bidirectional Encoder Representations from Transformers*) Google, 2018 – *маскированная* языковая модель для 104 разных ЕЯ
  - двунаправленная сеть на архитектуре *Transformer*
  - векторное представление целых фраз (матрица весов)
  - обучается на двух задачах:
    - *маскированное языковое моделирование*: предсказываются слова, замененные на [MASK]
    - предсказание следующего предложения (является ли вторая фраза продолжением первой)

# ПРИМЕНЕНИЕ НЕЙРОСЕТЕВЫХ ДИСТРИБУТИВНЫХ МОДЕЛЕЙ



- Модели уровня слов, статические эмбединги:
  - Исправление опечаток, неверной транслитерации слов, например, для *пщщпду*  
— *пщщщпду* - 0.723, ... *гугл* - 0.649, *поопду* - 0.647...
  - Построение классов семантически близких слов для разных приложений КЛ (тезаурусов) и многое другое
- Контекстуализированные эмбединги:
  - Обработка запросов к поисковику: запрос анализируется не как набор ключевых слов/фраз, а за счет векторного представления слов (тем самым учитывается весь контекст запроса, включая служебные слова)
  - Источник признаков для других обучаемых моделей КЛ, в частности, для классификации (*BERT*), с учетом или без «размораживания» внутренних весов модели



# ЗАКЛЮЧЕНИЕ



- Модели дистрибутивной семантики – активно развивающаяся область КЛ, наиболее применяемы:  
*Word2Vec, FastText, BERT, GPT (Generative Pre-trained Transformer, Open AI, 2020)*
- Модели имеют множество параметров, которые позволяют настроить, адаптировать их к конкретным прикладным задачам
- Развивается *трансферное обучение (transfer learning)*, “перенос знаний” и “тонкая настройка” (*fine tuning*) – для применения предобученной нейронной языковой модели для новых видов задач

## СПАСИБО ЗА ВНИМАНИЕ!