

# ДИСТРИБУТИВНАЯ СЕМАНТИКА

---

Вася Андриянец [bandrandr@yandex.ru](mailto:bandrandr@yandex.ru),  
Маша Шеянова, [masha.shejanova@gmail.com](mailto:masha.shejanova@gmail.com)

31.07.2019

ЛОШ КЛ

# INTRO

---

# что это?

Что мы хотим:

- уметь считать расстояние между словами
- учитывая только **значения слов** (насколько слова близки друг к другу по значению)
- делать это автоматически

Пример: **лампа** и **светильник** — ближе, чем **лампа** и **лавка**.

**Дистрибутивная гипотеза:** значения слов определяются их контекстами. Слова с похожими типичными контекстами имеют схожее значение.

You shall know a word by the company it keeps! (J.R.Firth)

# КАК ЭТО РАБОТАЕТ?

Нам нужно:

- много текстов, чтобы картинка была репрезентативной
- посчитать в этих текстах взаимную встречаемость слов друг с другом
- найти слова, которые могут заменить друг друга и слова, у которых нет общих контекстов

Готово! Мы прекрасны и можем

- находить слова, близкие по значению к данному
- строить семантические пропорции
- строить семантические визуализации

# ПРИМЕРЫ: RUSVECTORES

---

# ЧТО ТАКОЕ RUSVECTORES?

На rusvectors можно найти слова, наиболее близкие к данному, построить семантическую пропорцию и многое другое.

## Семантические аналоги для *спокойный* (ALL)

### НКРЯ и Wikipedia

1. невозмутимый 0.69
2. безмятежный 0.68
3. спокойный 0.67
4. -спокойный 0.66
5. несуетливый 0.65
6. умиротворенный 0.65
7. умиротворять 0.63
8. раздумчивый 0.63
9. неторопливый 0.62
10. кроткий 0.62

### Новостной корпус

1. умиротворенный 0.52
2. размеренный 0.50
3. безмятежный 0.50
4. беспокойный 0.50
5. уравновешенный 0.49
6. расслабленный 0.47
7. беспокойный 0.47
8. неторопливый 0.45
9. доброжелательный 0.45
10. дружелюбный 0.44



человек\_S



нога\_S

## News corpus

1. ступня 0.430
2. котенок 0.424
3. кошачий 0.409
4. пес 0.403
5. ножка 0.388

## Web corpus

1. лапа 0.534
2. ступня 0.519
3. колено 0.508
4. спина 0.484
5. туловище 0.472

кошка\_S



???

## Ruscorpora

1. лапка 0.499
2. ножка 0.485
3. лапа 0.482
4. ножища 0.482
5. ножонка 0.479

-

Choose the model:

☒ Ruscorpora and Russian Wikipedia

Show only results which belong to:

☐ Nouns ☐ Verbs ☐ Adverbs

Calculate!

Choose the model:

☒ Ruscorpora and Russian Wikipedia ☒ News corpus ☒ Ruscorpora ☒ Web corpus

## Новостной корпус

Визуализировать в TensorFlow Projector



# КАК ЭТО РАБОТАЕТ

---

Во введении я говорила, что мы считаем близость на основе контекстов. Но как именно?

Превращаем слова в **векторы**  
и измеряем **косинусное расстояние**<sup>1</sup> между ними.

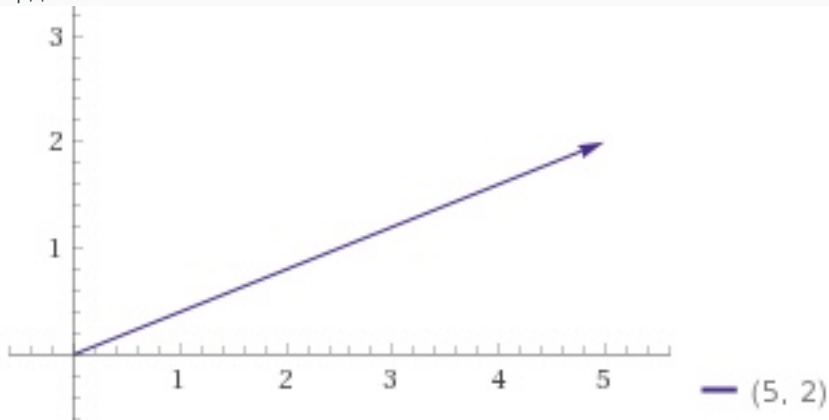
---

<sup>1</sup>Есть, впрочем, и другие метрики.

ЧТО ТАКОЕ ВЕКТОР?

# ЧТО ТАКОЕ ВЕКТОР

В школе нас обычно учат, что вектор — это стрелочка в системе координат.



Ту же стрелочку можно представить как набор чисел:  $(5, 2)$ .

А что если пространство 3-мерное? 4-мерное? 100500-мерное? В 3D стрелочку представить ещё можно. А в 100500-мерном придётся обходиться числами.

Но **как** мы векторизуем слово? Как уже было сказано, по контекстам.



## МОЖНО ЛИ ЭТО НАСТРАИВАТЬ?

Да. Например, можно поиграться с размером окна.

Можно считать все вхождения слов в окне от 5 до нашего слова до 5 после:

туда [пришла. Потом мы начали смотреть **мультки** и до двух  
ночи не] ложились ...

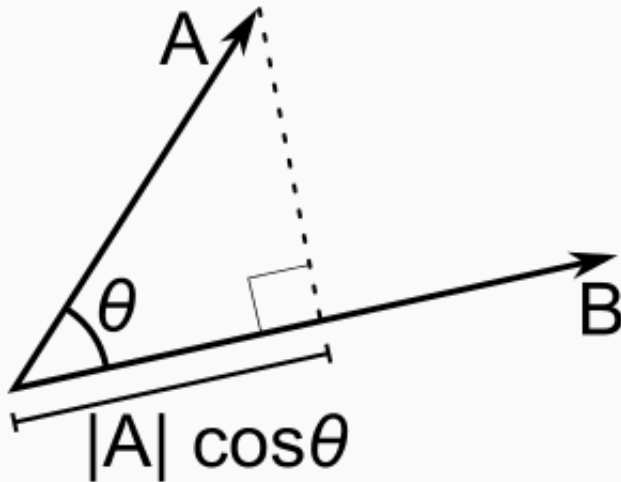
А можно — от -3 до +3:

туда пришла. Потом [мы начали смотреть **мультки** и до двух]  
ночи не ложились ...

От этого будет зависеть, в каких отношениях находятся близкие, согласно нашей модели, слова.

## КОСИНУСНАЯ БЛИЗОСТЬ

Что это? Да просто косинус угла между векторами!



Чем косинус угла ближе к единице, тем ближе слова друг к другу, чем ближе к 0 — тем дальше.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

У счётных моделей есть глобальные недостатки:

- Размер векторов получается огромным (в общем случае равен объёму лексикона).
- Это очень замедляет операции сравнения векторов.
- Мы не знаем точно, что в наших векторах нужная информация, а что мусор. Они просто взяты из корпуса.

Как быть? Используем нейросеточки для **предсказания** векторов.

Мы пытаемся для каждого слова найти такой вектор, чтобы он был **максимально схож** с векторами типичных соседей и **максимально отличался** от векторов слов, которые соседями данному слову не являются.

# ИНСТРУМЕНТЫ

---

В 2013 году исследователь Tomas Mikolov из Google с соавторами разработали word2vec, который позволяет тренировать нейронные языковые модели на больших корпусах.

Сейчас для многих языков (например, для русского) есть готовые обученные модели модели!

А что если нам надо векторизовать **целый текст**?

Самый простой вариант — сделать это с помощью усреднения векторов каждого слова в документе.

Цитата из статьи про Doc2VecC:

In Doc2VecC, we represent each document as a simple average of the word embeddings of all the words in the document.

А можно не просто складывать вектора слов, а домножать их на разные **веса**, чтобы одни слова вносили больший вклад, чем другие.



# ПРИМЕНЕНИЕ

---

ОКЕЙ, МЫ УМЕЕМ СЧИТАТЬ РАССТОЯНИЕ  
МЕЖДУ СЛОВАМИ. ЧТО ДАЛЬШЕ?

- поиск синонимов и вообще похожих слов
- снятие семантической омонимии (Word Sense Disambiguation)
- признаки для машинного обучения с текстами
- ... и туча всего другого!

СПАСИБО ЗА ВНИМАНИЕ!  
Вопросы?