

Современные дистрибутивно-семантические модели и их применение в лингвистических исследованиях

Андрей Кутузов
Университет Осло
Группа лингвистических технологий

Школа компьютерной лингвистики
Тюмень, февраль 2018



- 1 Что такое компьютерная лингвистика?
- 2 Дистрибутивная семантика
 - Демо-сервис для примера
 - Дистрибутивная гипотеза
 - Векторные модели
 - Сжатые вектора (word embeddings)
 - Счетные дистрибутивные модели
 - Предсказательные дистрибутивные модели: Word2Vec
- 3 Практические аспекты



Много названий:

- ▶ Computational Linguistics (CL);
- ▶ Natural Language Processing (NLP);
- ▶ Natural Language Understanding (NLU);
- ▶ Более или менее одна и та же академическая дисциплина:
 - ▶ научное изучение языка с вычислительной точки зрения.

История

- ▶ Ещё средневековые мистики искали закономерности в священных текстах;
- ▶ Но в современном понимании наша наука начинается в XX веке:
 - ▶ Джордж Ципф (изучал статистику естественных языков);
 - ▶ Ноам Хомский (изобрёл трансформационную грамматику);
 - ▶ бум машинного перевода в 1950-е годы.



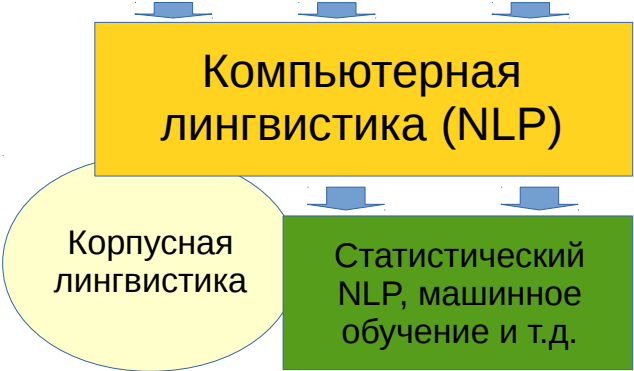
Количество докладов на ежегодной конференции Ассоциации
Компьютерной Лингвистики (ACL)

- ▶ Рост интереса к NLP последние 20 лет: почему?
- ▶ **Информация**: ‘нефть XXI века’;
- ▶ Все хотят обрабатывать потоки информации (особенно IT-компании);
 - ▶ Информация очень часто содержится в (оцифрованных) **текстах**.
- ▶ Важно: NLP это и **академическая** и **индустриальная** (прикладная) наука!
 - ▶ исследователи легко перемещаются из университетов в коммерческие компании и обратно.
- ▶ Компьютерные лингвисты участвуют в разработке многих современных систем:
 - ▶ машинный перевод
 - ▶ распознавание речи
 - ▶ интернет-поисковики
 - ▶ проверка орфографии и грамматики
 - ▶ виртуальные личные помощники (Siri, Alexa, Cortana)
 - ▶ ...и в конечном итоге - **искусственный интеллект** (artificial intelligence).

Отличия от 'традиционной' или 'общей' лингвистики

- ▶ Традиционная лингвистика **описывает и сравнивает языки**.
- ▶ NLP ближе к математике и инженерным наукам: мы **вычисляем**.
- ▶ Построение **вычислительных моделей лингвистических явлений**:
 1. **'правилые'** ('rule-based');
 2. **'статистические'** ('data-based').
- ▶ Статистика лежит в основе современного NLP.
- ▶ Мы проводим **эксперименты** для проверки **гипотез**:
 - ▶ 'в этом языке 10 частей речи',
 - ▶ 'информация о совместной встречаемости слов улучшает качество классификации документов'.
- ▶ **Replicability** (один и тот же эксперимент должен выдавать идентичные результаты);
- ▶ **Reproducibility** (похожие эксперименты должны выдавать сравнимые результаты).

Общая лингвистика



```
graph TD; A[Общая лингвистика] --> B[Компьютерная лингвистика (NLP)]; A --> C[Корпусная лингвистика]; A --> D[Статистический NLP, машинное обучение и т.д.]; B --> D; C --> D;
```

Компьютерная
лингвистика (NLP)

Корпусная
лингвистика

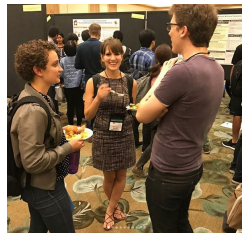
Статистический
NLP, машинное
обучение и т.д.



- ▶ Исследование должно быть **практическим**.
- ▶ ‘Покажи свой код!’
- ▶ ‘Покажи качество своей системы!’
- ▶ **Эмпирическая оценка** (evaluation) на конкретных проблемах.
- ▶ **Тестовые датасеты**.
- ▶ **Открытые соревнования** (shared tasks).

► Конференции:

- ACL
- EMNLP
- EACL
- NAACL
- COLING
- LREC...

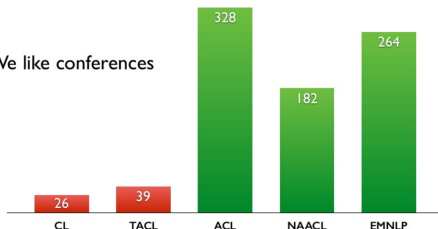


► Журналы:

- ‘Computational Linguistics’ (CL);
- ‘Transactions of the Assoc. for Computational Linguistics’ (TACL).

► В отличие от многих других наук, **журналы менее важны**.

• We like conferences



- ▶ Большинство статей можно свободно скачать в **Association for Computational Linguistics (ACL) Anthology**:
 - ▶ <https://aclanthology.info/>
- ▶ Практически везде работает **двойное слепое рецензирование...**
- ▶ ...в последние годы также становится популярной открытая публикация на препринт-серверах:
 - ▶ <https://arxiv.org/list/cs.CL/recent>

[most recent](#) | [top recent](#) | [top hype](#) | [friends](#) | [discussions](#) | [recommended](#) | [library](#)

Building a Conversational Agent Overnight with Dialogue Self-Play
Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bagpa, Neha Nayak, Larry Heck
1/15/2018 [cs.AI](#) | [cs.CL](#)
11 pages, 4 figures

1801.04871v1 [pdf](#)
[show similar](#) | [discuss](#)



We propose Machines Talking To Machines (M2M), a framework combining automation and crowdsourcing to rapidly bootstrap end-to-end dialogue agents for goal-oriented dialogues in arbitrary domains. M2M scales to new tasks with just a task schema and an API client from the dialogue system developer, but it is also customizable to cater to task-specific interactions. Compared to the Wizard-of-Oz approach for data collection, M2M achieves greater diversity and coverage of salient dialogue flows while maintaining the naturalness of individual utterances. In the first phase, a simulated user bot and a domain-agnostic system bot converse to exhaustively generate dialogue "outlines", i.e. sequences of template utterances and their semantic parses. In the second phase, crowd workers provide contextual rewrites of the dialogues to make the utterances more natural while preserving their meaning. The entire process can finish within a few hours. We propose a new corpus of 3,000 dialogues spanning 2 domains collected with M2M, and present comparisons with popular dialogue datasets on the quality and diversity of the surface forms and dialogue flows.

Predicting Movie Genres Based on Plot Summaries

Quen Hoang
1/15/2018 [cs.CL](#) | [cs.LG](#) | [stat.ML](#)

1801.04813v1 [pdf](#)
[show similar](#) | [discuss](#)



This project explores several Machine Learning methods to predict movie genres based on plot summaries. Naive Bayes, Word2Vec+XGBoost and Recurrent Neural Networks are used for text classification, while K-binary transformation, rank method and probabilistic classification with learned probability threshold are employed for the multi-label problem involved in the genre tagging task. Experiments with more than 250,000 movies show that employing the Gated Recurrent Units (GRU) neural networks for the probabilistic classification with learned probability threshold approach achieves the best result on the test set. The model attains a Jaccard Index of 50.0%, a F-score of 0.56, and a hit rate of 50.5%.

- ▶ Сейчас на NLP активно влияют другие науки:
- ▶ **data science** и **машинное обучение** (machine learning).
 - ▶ Некоторые проблемы настолько сложны, что мы **не можем сформулировать точные алгоритмы их решения**.
 - ▶ В таких случаях используется **машинное обучение**:
 - ▶ программы, которые **учатся** делать правильные решения на некотором обучающем материале;
 - ▶ итак, мы **обучаем** наши системы на языковых данных (обычно на больших собраниях текстов, корпусах).
 - ▶ **Искусственные нейронные сети** - это один из популярных алгоритмов машинного обучения для моделирования языка.

Возрождение глубокого обучения

- ▶ '**Глубокое обучение**' (deep learning) - это обучение моделей с использованием многослойных искусственных нейронных сетей.
- ▶ После долгого 'застоя' 60-х, 70-х и 80-х годов, оно снова популярно.
- ▶ Глубокое обучение оказалось очень эффективным в NLP-задачах.
- ▶ 'Нужно ли нам вообще что-то кроме нейронных сетей?'
- ▶ И это ещё одна причина роста интереса к нашей науке.

- ▶ NLP: **наука** или **инженерная дисциплина**?
- ▶ Возможно, **CL** это наука, а **NLP** это её применение к **эмпирическим проблемам**?
- ▶ Люди занимаются исследованиями по разным причинам:
 1. ...пытаясь найти **вычислительное объяснение лингвистическому или психолингвистическому явлению**;
 2. ...пытаясь создать **рабочий компонент прикладной системы для работы с языком**.
- ▶ К чему тогда относятся топовые конференции: к **CL** или к **NLP**?
- ▶ Подавляющее большинство статей сконцентрированы на практических вопросах.
- ▶ Окончательного ответа нет.



Yoav Goldberg

Follow

Senior Lecturer at Bar Ilan University. Working on NLP. Recently with Neural Nets. Published a book about it. <http://www.cs.biu.ac.il/~yogo/>

Jun 9, 2017 · 14 min read

An Adversarial Review of "Adversarial Generation of Natural Language"

Or, for fucks sake, DL people, leave language alone and stop saying you solve it.

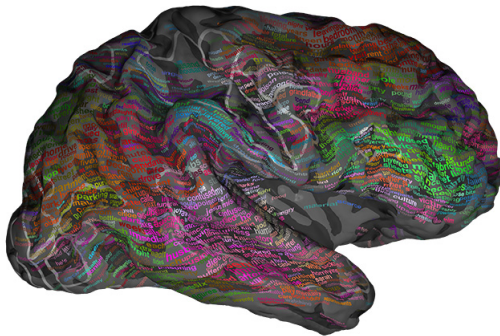
‘...человеческий язык огромен, сложен и труден. В нём множество нюансов, странностей и удивительных исключений. Хотя NLP-исследователи —и лингвисты, которые делают самую тяжёлую работу!—добились многого в понимании языка и того, как его следует обрабатывать, тем не менее, **мы всё ещё в самом начале очень долгого пути.**’

Возвращение языкознания

- ▶ NLP снова обращается к **лингвистической структуре**;
 - ▶ Это признают даже самые яростные сторонники чисто статистических подходов;
 - ▶ Лингвистические структуры, встраиваемые в алгоритмы машинного обучения радикально сокращают пространство поиска и в итоге сильно улучшают качество [Dyer, 2017];
 - ▶ **Язык – это не просто последовательность слов / символов / байтов.**
-
- ▶ Но **что NLP может дать традиционной лингвистике?**
 - ▶ И вообще гуманитарным наукам?
 - ▶ Давайте перейдём к конкретике.

- 1 Что такое компьютерная лингвистика?
- 2 Дистрибутивная семантика
 - Демо-сервис для примера
 - Дистрибутивная гипотеза
 - Векторные модели
 - Сжатые вектора (word embeddings)
 - Счетные дистрибутивные модели
 - Предсказательные дистрибутивные модели: Word2Vec
- 3 Практические аспекты

- ▶ Как интернет-поисковики находят релевантные документы, даже если в них нет ни одного слова из запроса?
- ▶ Как системы машинного перевода выбирают правильный перевод для неизвестных им слов?
- ▶ Как Siri, Alexa и Cortana, etc) определяют эмоцию высказывания, хотя их не программировали специально на эти слова?
- ▶ Кажется, будто эти системы каким-то образом ‘знают значение’ слов, которые они обрабатывают...
- ▶ ...как мы, люди.



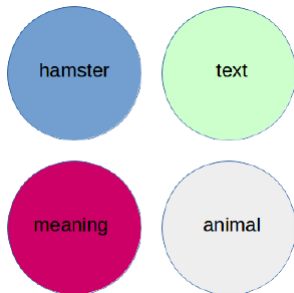
Нам нужна **машина**, имитирующая человеческий мозг и **понимающая значение слова**.

Как построить такую машину?

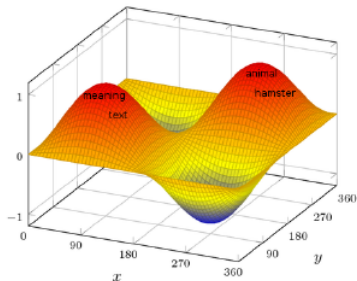
- ▶ Я буду говорить о **векторных моделях значения...**
- ▶ ...они же **‘дистрибутивные модели’**.
- ▶ Не так давно они практически полностью покорили компьютерную лингвистику.
- ▶ Они используются во множестве академических и промышленных проектов.
- ▶ Почему?
- ▶ Потому что они эффективно **вычисляют семантическую близость между лингвистическими сущностями.**
- ▶ Например, словами.

Дискретные (независимые) репрезентации слов – это старая проблема в NLP.

Дискретные репрезентации слов
(плохо)



Непрерывные репрезентации слов
(хорошо)



Мы бы хотели, чтобы слова были представлены какими-то осмысленными ‘**координатами в семантическом пространстве**’.

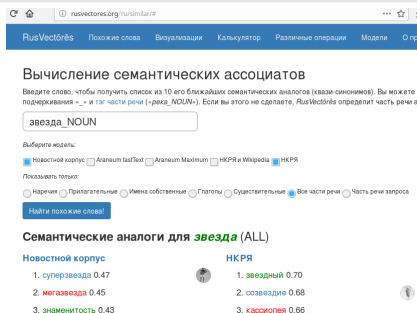
Дистрибутивно-семантические модели для русского языка в Интернете

Сайт Национального Корпуса Русского языка (НКРЯ):

<http://ruscorpora.ru>.

Наш веб-сервис: **RusVectōrēs** ('русские векторы' на латыни)

<http://rusvectors.org>



The screenshot shows the RusVectores website interface. At the top, there's a navigation bar with links: RusVectōrēs, Похожие слова, Визуализации, Калькулятор, Различные операции, Модели, and О проекте. The main heading is "Вычисление семантических ассоциатов". Below it, a text box contains the input "звезда_NOUN". Underneath, there are checkboxes for selecting models: "Новостной корпус", "Araneum tasPlex", "Araneum Maximum", "НКРЯ и Wikipedia", and "RusVectores". There are also checkboxes for "Показывать только:" with options like "Наречия", "Прилагательные", "Имена собственные", "Глаголы", "Существительные", "Все части речи", and "Часть речи запроса". A blue button says "Найти похожие слова!". Below this, the section "Семантические аналоги для звезда (ALL)" is shown, divided into two columns: "Новостной корпус" and "НКРЯ". The "Новостной корпус" column lists: 1. суперзвезда 0.47, 2. мегазвезда 0.45, 3. знаменитость 0.43. The "НКРЯ" column lists: 1. звездный 0.70, 2. созвездие 0.68, 3. космолет 0.66.

(работает и на мобильных устройствах)

Уровни лингвистического анализа

Компьютерная лингвистика сравнительно легко моделирует нижние уровни языка:

- ▶ **графематику** – как слова пишутся
- ▶ **фонетику** – как слова произносятся
- ▶ **морфологию** – как слова склоняются и изменяются
- ▶ **синтаксис** – как слова сочетаются в предложении

Дистрибутивная гипотеза

Моделировать значит в сжатом виде представить важные черты явления. Например, во фразе ‘The judge sits in the court’, слово ‘judge’:

1. состоит из 3 фонем [j e j];
2. является **существительным единственного числа в именительном падеже**;
3. выступает как **подлежащее** в синтаксическом дереве предложения.

Такие **локальные репрезентации** говорят нам многое о слове ‘judge’. Но не о его значении.

Как представить значение?

- ▶ Семантика сопротивляется формальному представлению.
- ▶ Нам нужны машино-читаемые репрезентации слов.
- ▶ Слова, похожие по значению должны обладать математически похожими репрезентациями.
- ▶ ‘Светильник’ похож на ‘лампу’, но не на ‘кипятильник’, хотя поверхностная форма слов говорит о противоположном.
- ▶ Почему так?

Дистрибутивная гипотеза

Произвольность языкового знака

- ▶ В отличие от **дорожных знаков**, у слов нет прямой связи между формой и значением.
- ▶ Это известно ещё со времен **Фердинанда де Соссюра** (структурализм вообще сильно повлиял на дистрибутивную семантику).
- ▶ Концепт 'светильник' можно выразить любой поверхностной формой:



▶ **lantern**

▶ **лампа**

▶ **лампа**

▶ **lucerna**

▶ **гэрэл**

▶ ...

Дистрибутивная гипотеза

Откуда мы знаем, что у слов ‘лампа’ и ‘светильник’ похожие значения? И что такое вообще **значение**?

И как заставить компьютеры понимать всё это?

Возможные источники данных

Методы вычислительной репрезентации семантики в естественных языках:

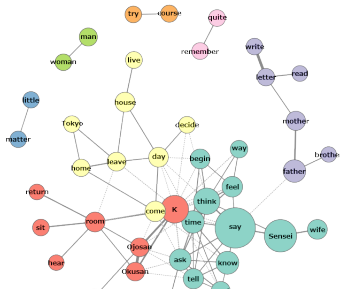
1. **Ручное построение онтологий** (подход от знаний). Работает сверху вниз: от абстракций к реальным текстам. Например, **Wordnet**.
2. **Извлечение семантики из статистики больших корпусов** (дистрибутивный подход). Работает снизу вверх: от реальных текстов к абстракциям. Например, **word2vec**.

Тема нашего трека на этой школе – второй подход

Дистрибутивная гипотеза

Дистрибутивная гипотеза

- ▶ Значение – это сумма контекстов
- ▶ У слов с типично похожими контекстами - похожее значение
- ▶ Впервые сформулирована у Людвиг Витгенштейна (1930-е) и у [Harris, 1954].
- ▶ ‘You shall know a word by the company it keeps’ [Firth, 1957]
- ▶ Дистрибутивно-семантические модели (DSM) строятся на данных о совместной встречаемости слов в больших корпусах.



Дистрибутивная гипотеза

Важно различать **синтагматические** и **парадигматические** отношения между словами.

- ▶ Слова находятся в **синтагматических** отношениях, если они типично стоят рядом в тексте ('есть хлеб'). Также называется **совместная встречаемость первого порядка**.
- ▶ **Синтагма** выступает как **упорядоченный список**.
- ▶ Слова находятся в **парадигматических** отношениях, если возле них типично встречаются одни и те же соседи (мы часто 'едем' как 'хлеб', так и 'масло'). Также называется **совместная встречаемость второго порядка**.
- ▶ **Парадигма** – это скорее **массив взаимозаменяемых элементов**.

Нас в основном интересуют **парадигматические** отношения ('хлеб' семантически близок к 'маслу', но не к 'свежий').



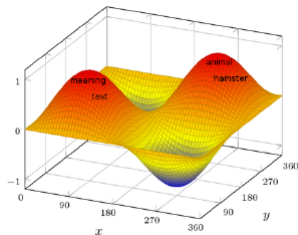
Основной метод представления значения в дистрибутивной семантике – **семантические вектора**.

Их изобрел американский психолог **Чарльз Осгуд** в 1950-е годы [Osgood et al., 1964].

Почему вектора?

Что такое вектор?

- ▶ Вектор – это просто последовательность из n чисел:
 - ▶ $[0, 1, 2, 4]$ это вектор с 4 компонентами;
 - ▶ $[20, 89, 34]$ это вектор с 3 компонентами;
- ▶ Можно считать, что компоненты – это координаты в n -мерном пространстве;
- ▶ тогда вектор – это точка в этом пространстве.
- ▶ 3-мерное пространство:



Почему вектора?

Базируясь на обучающем корпусе, мы можем сгенерировать вектора для каждого слова в нем.

Вектора слов

- ▶ Каждое слово A представлено вектором или точкой \vec{A} в многомерном семантическом пространстве;
- ▶ его компоненты или координаты – это другие слова из словаря обучающего корпуса ($B, C, D \dots N$);
- ▶ значения компонентов – частоты совместной встречаемости с этими другими словами в нашем корпусе.

Дистрибутивная семантика

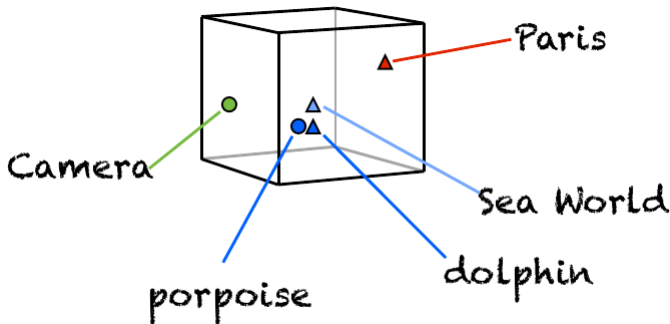
- ▶ Представим простой текст, упоминающий ‘хомяков’, ‘вектора’, ‘значения’, ‘корпуса’, ‘животных’ и ‘сусликов’.
- ▶ Можно посчитать, как часто эти слова стоят рядом друг с другом в этом тексте.
- ▶ Получится простая симметричная **матрица совместной встречаемости**:

| | вектор | значение | хомяк | корпус | суслик | животное |
|----------|--------|----------|-------|--------|--------|----------|
| вектор | 0 | 10 | 0 | 8 | 0 | 0 |
| значение | 10 | 0 | 1 | 15 | 0 | 0 |
| хомяк | 0 | 1 | 0 | 0 | 20 | 14 |
| корпус | 8 | 15 | 0 | 0 | 0 | 2 |
| суслик | 0 | 0 | 20 | 0 | 0 | 21 |
| животное | 0 | 0 | 14 | 2 | 21 | 0 |

У семантически похожих слов похожие 6-D вектора (строки).

Дистрибутивная семантика

Похожие слова близки друг другу в пространстве их типичных контекстов



Дистрибутивная модель – это просто список слов и соответствующих им векторов.

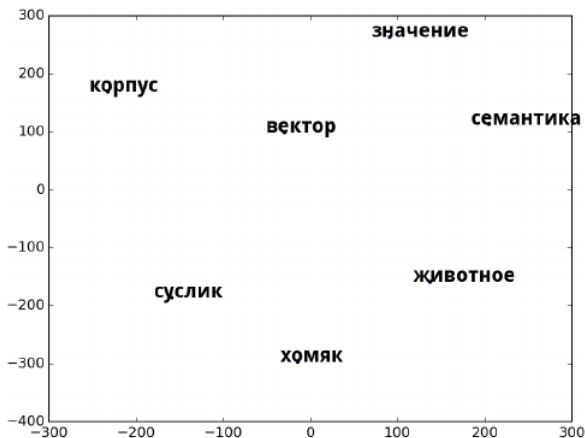
Сжатые вектора (word embeddings)

Проклятие размерности

- ▶ С большими корпусами мы получим **миллионы измерений** (осей, контекстов).
- ▶ Но вектора очень **разреженные**, большинство компонентов равны нулю.
- ▶ Разными математическими трюками можно **снизить размерность векторов** до разумных значений...
- ▶ ..и всё ещё сохранить значимые отношения между ними
- ▶ Такие сжатые вектора называются **‘word embeddings’**.

Сжатые вектора (word embeddings)

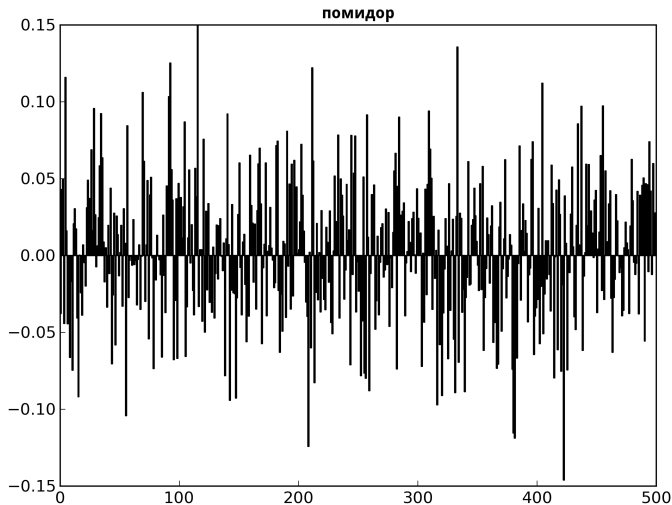
Размерность можно сокращать, пока не дойдем до 2 или 3 компонентов. 2-мерное или 3-мерное пространство уже можно визуализировать.



Вектора, сжатые до 2 компонентов и визуализированные в t-SNE

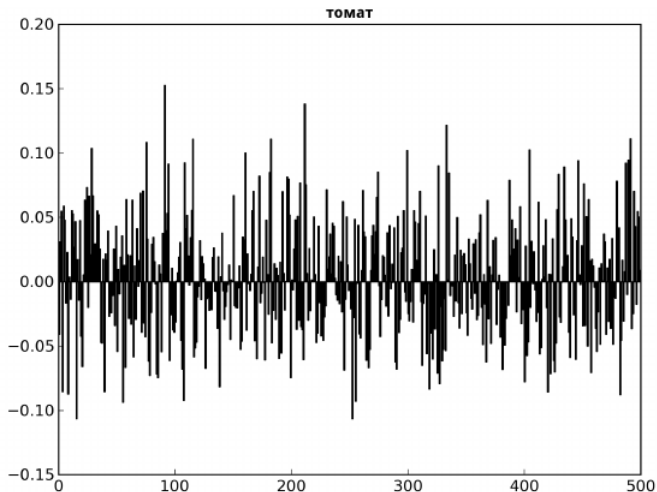
Сжатые вектора (word embeddings)

500-D вектор для 'помидор'



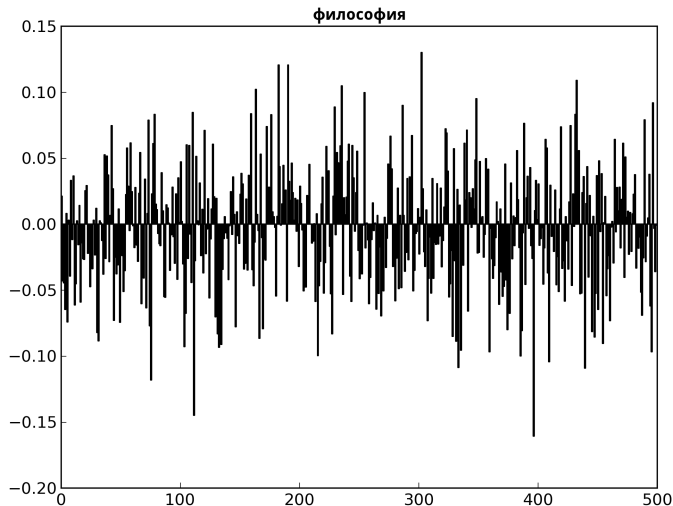
Сжатые вектора (word embeddings)

500-D вектор для 'томат'



Сжатые вектора (word embeddings)

500-D вектор для 'философия'



Можем ли мы доказать, что **томат** ближе к **помидору**, чем к **философии**?

Сжатые вектора (word embeddings)

Семантическая близость между словами измеряется через **косинус** угла между их векторами (принимает значения от -1 to 1).

- ▶ Схожесть падает, **когда угол увеличивается**.
- ▶ Схожесть растёт, **когда угол уменьшается**.

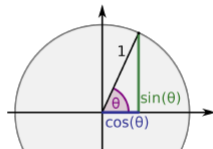
$$\cos(w1, w2) = \frac{\vec{V}(w1) \times \vec{V}(w2)}{|\vec{V}(w1)| \times |\vec{V}(w2)|} \quad (1)$$

(скалярное произведение нормализованных векторов)

Куда указывают вектора?

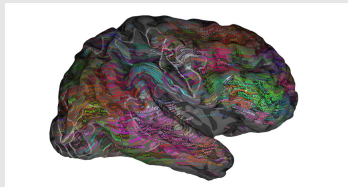
- ▶ Вектора указывают в одном направлении: косинус = 1;
- ▶ Вектора перпендикулярны (ортогональны): косинус = 0;
- ▶ Вектора указывают в противоположных направлениях: косинус = -1.

$\cos(\text{tomat}, \text{philosophy}) = 0.00698$
 $\cos(\text{pomidor}, \text{philosophy}) = -0.03429$
 $\cos(\text{tomat}, \text{pomidor}) = 0.65049$



Ближайшие семантические соседи (ассоциаты)

МОЗГ



1. гиппокамп 0.61
2. нейрон 0.61
3. психика 0.58
4. мозжечок 0.58
5. ...

Сжатые вектора (word embeddings)

Можно получить вектора и для словосочетаний



Боб Дилан

1. эрик клэптон 0.79
2. брюс спрингстин 0.73
3. пол маккартень 0.72
4. джордж харрисон 0.72
5. джим моррисон 0.72
6. ...

Попробуйте сами на <http://rusvectors.org>!

Сжатые вектора (word embeddings)

Основные подходы к конструированию word embeddings

1. Матрицы совместной встречаемости, факторизованные через **SVD** (так называемые **счетные модели**) [Bullinaria and Levy, 2007];
2. **Предсказательные модели**, использующие простые **искусственные нейронные сети**, впервые описаны в [Bengio et al., 2003] и в [Mikolov et al., 2013] (**word2vec**).

Предсказательные модели сейчас переживают невиданный взлет и используются почти во всех областях NLP.

Их принципиальное отличие: использование **машинного обучения**.

Счетные дистрибутивные модели

Традиционные дистрибутивные модели (счетные)

Алгоритм построения счетной модели

1. посчитать полную матрицу совместной встречаемости на всём корпусе;
2. взвесить абсолютные частоты при помощи меры positive point-wise mutual information (PPMI);
3. факторизовать эту матрицу при помощи сингулярного разложения (SVD), чтобы снизить размерность и превратить разреженные вектора в сжатые.

Подробности в [Bullinaria and Levy, 2007] и в описаниях методов типа Latent Semantic Indexing (LSI) или Latent Semantic Analysis (LSA).

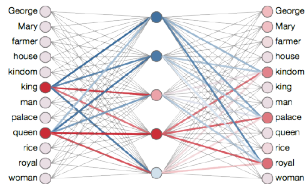
- ▶ Дистрибутивные модели на основе машинного обучения часто называются ‘предсказательные модели’.
- ▶ В **счетных моделях** мы считаем частоты совместной встречаемости слов и используем их как вектора;
- ▶ в **предсказательных моделях** всё наоборот:
- ▶ мы пытаемся найти (выучить) для каждого слова такой вектор (embedding), чтобы он был **максимально похож** на вектора его **парадигматических соседей**
- ▶ ...и **минимально похож** на вектора слов, которые в этом корпусе не являются **парадигматическими соседями** данного слова.

Если используются нейронные сети, то такие выученные вектора называются **neural embeddings**.

Предсказательные дистрибутивные модели: Word2Vec

Как работает мозг

В нашем мозге 10^{11} нейронов, и у каждого 10^4 соединений. Нейроны получают от других нейронов сигналы разной интенсивности. В зависимости от полученного сигнала, они отправляют дальше по сети разные реакции.



Векторные модели семантики пытаются имитировать этот процесс.

Нейроны и вектора

Возможно, концепты хранятся в мозгу как **паттерны активации нейронов**.

Это очень похоже на векторные репрезентации! Значение – это массив распределенных **‘семантических компонентов’**; каждый из них может быть более или менее активирован (выражен).



Концепты представлены векторами из n компонентов (или **нейронов**), и каждый **нейрон** отвечает за несколько концептов или **‘семантических компонентов’**.

Предсказательные дистрибутивные модели: Word2Vec



- ▶ В 2013 году Томаш Миколов из Google опубликовал статью ‘Efficient Estimation of Word Representations in Vector Space’;
- ▶ кроме того, был опубликован исходный код программы **word2vec**, реализующей описанные в статье алгоритмы...
- ▶ ...и дистрибутивная модель, обученная на большом корпусе Google News.
 - ▶ [Mikolov et al., 2013]
 - ▶ <https://code.google.com/p/word2vec/>

- ▶ До этого Bengio и Collobert уже обучали сложные нейронные сети для **предсказания следующего слова в предложении** [Bengio et al., 2003];
- ▶ вектора (эмбединги) для отдельных слов получались как побочный эффект обучения.
- ▶ Миколов слегка модифицировал эти алгоритмы;
- ▶ сделал **выучивание хороших лексических векторов** основной целью модели.
- ▶ word2vec оказался чрезвычайно быстрым и эффективным.
- ▶ NB: на самом деле в нем **два** разных алгоритма:
 - ▶ **Continuous Bag-of-Words (CBOW)**
 - ▶ **Continuous Skipgram**.

...an efficient method for learning high quality distributed vector ...

context focus word context

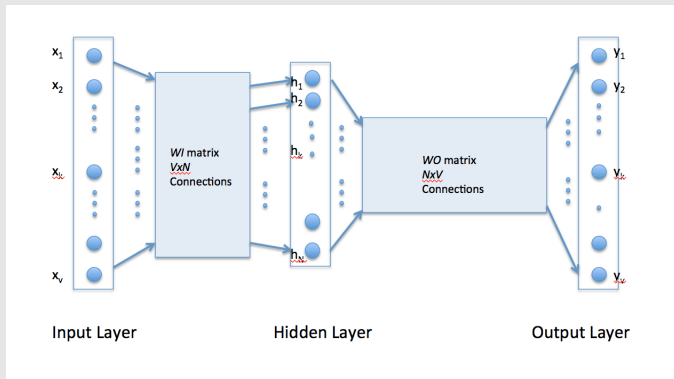
Обучение векторов

- ▶ Сначала **генерируются случайные вектора** для каждого слова.
- ▶ Во время обучения, мы движемся по корпусу **скользящим окном**.
- ▶ Каждое слово в тексте воспринимается как задача предсказания:
- ▶ мы хотим **предсказать это слово при помощи слов-соседей**.
- ▶ NB: 'предсказать' тут означает 'найти у какого слова в словаре самая высокая **косинусная близость** с усредненным вектором слов-контекстов'.
- ▶ Исход предсказания (правильное или нет) определяет, **сдвигаем** ли мы вектор текущего слова и в каком направлении.
- ▶ Постепенно вектора приходят в оптимальное состояние.

Предсказательные дистрибутивные модели: Word2Vec

Обучающие алгоритмы CBOW и SkipGram

‘вектор слова **w** ‘раскачивается’ туда-сюда векторами его контекстов, как если бы между ними были материальные струны или канаты...это как гравитация.’ [Rong, 2014]



Очень хорошая визуальная демонстрация алгоритмов word2vec:
<https://ronxin.github.io/wevi/>

Предсказательные дистрибутивные модели: Word2Vec

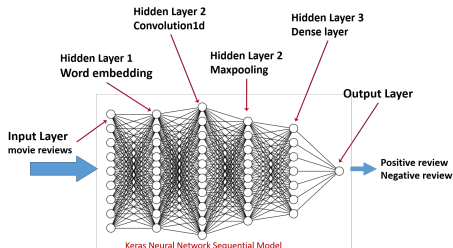
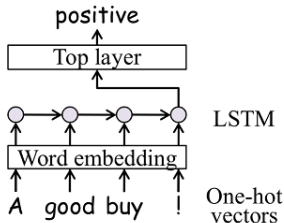
В течение двух лет после статьи Миколова появилось множество последователей:

- ▶ Christopher Manning и другие исследователи из Стэнфорда опубликовали **GloVe** – видоизмененную версию похожего подхода [Pennington et al., 2014];
- ▶ Omer Levy и Yoav Goldberg показали, что **SkipGram** на самом деле внутри тоже факторизует матрицу PMI-коэффициентов [Levy and Goldberg, 2014];
- ▶ Le и Mikolov предложили **Paragraph Vector**: алгоритм, выучивающий векторные репрезентации не только для слов, но и для абзацев или документов [Le and Mikolov, 2014];
- ▶ Миколов перешел на работу в Facebook и придумал **fastText**: алгоритм, который выучивает вектора для последовательностей символов (n-грамм) [Bojanowski et al., 2017];
- ▶ Все эти алгоритмы были реализованы в свободных и открытых библиотеках, например, в **Gensim** и **TensorFlow**.

Предсказательные дистрибутивные модели: Word2Vec

Сегодня сжатые вектора широко используются вместо дискретных репрезентаций лексики в качестве **входных данных для более сложных нейронных моделей**:

- ▶ feedforward networks,
- ▶ convolutional networks,
- ▶ recurrent networks,
- ▶ LSTMs...



- 1 Что такое компьютерная лингвистика?
- 2 Дистрибутивная семантика
 - Демо-сервис для примера
 - Дистрибутивная гипотеза
 - Векторные модели
 - Сжатые вектора (word embeddings)
 - Счетные дистрибутивные модели
 - Предсказательные дистрибутивные модели: Word2Vec
- 3 Практические аспекты

Основное программное обеспечение

1. Dissect [Dinu et al., 2013]
(<http://clic.cimec.unitn.it/composes/toolkit/>);
2. Исходный C-код **word2vec** [Le and Mikolov, 2014]
(<https://word2vec.googlecode.com/svn/trunk/>)
3. Библиотека Gensim для Python, включает реализации алгоритмов **word2vec**, **fastText** и других
(<https://github.com/RaRe-Technologies/gensim>);
4. Реализация **word2vec** в TensorFlow от Google
(<https://www.tensorflow.org/tutorials/word2vec>);
5. Реализация **GloVe** от его авторов [Pennington et al., 2014]
(<http://nlp.stanford.edu/projects/glove/>).

Вы можете найти модели в разных форматах

1. Простой **текстовый формат**: слова и последовательности чисел – их вектора, одно слово на строку; в первой строке даётся информация о числе слов в модели и размерности вектора.
2. То же самое в **бинарной форме**.
3. **Бинарный формат Gensim**: использует NumPy-матрицы, сохранённые методами Python; хранит много дополнительной информации о модели.

Gensim работает со всеми этими форматами.

Что нужно помнить о дистрибутивных моделях?



- ▶ основаны на **дистрибуции** (распределении) слов в больших обучающих корпусах;
- ▶ представляют значение слова в виде сжатого вектора (**word embedding**);
- ▶ слова, встречающиеся в похожих контекстах, получают **похожие вектора**;
- ▶ векторные представления **непрерывны** (continuous):
 - ▶ слова находятся в общем векторном пространстве и могут быть **ближе или дальше друг от друга**.
- ▶ Можно находить ближайших **семантических соседей** данного слова через вычисление **косинусной близости** между векторами.
- ▶ Это **парадигматическая** близость.



- ▶ Работаем с **Gensim** и моделями с **RusVectores**.
- ▶ Используем для этого **Python 3**.
- ▶ Полезный код:
 - ▶ <https://github.com/elmiram/2016learnpython/blob/master/word2vec.ipynb>
- ▶ Репозиторий со слайдами и кодом по нашему треку:
 - ▶ https://github.com/akutuzov/utmn_school_2018

Спасибо за внимание!
Вопросы?

Современные дистрибутивно-семантические модели
и их применение в лингвистических исследованиях

<http://rusvectors.org>

Андрей Кутузов (andreku@ifi.uio.no)
Language Technology Group
University of Oslo



Bengio, Y., Ducharme, R., and Vincent, P. (2003).
A neural probabilistic language model.
Journal of Machine Learning Research, 3:1137–1155.



Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017).
Enriching word vectors with subword information.
Transactions of the Association of Computational Linguistics,
5:135–146.



Bullinaria, J. A. and Levy, J. P. (2007).
Extracting semantic representations from word co-occurrence
statistics: A computational study.
Behavior research methods, 39(3):510–526.



Dinu, G., Pham, N. T., and Baroni, M. (2013).
Dissect - distributional semantics composition toolkit.
In Proceedings of the 51st Annual Meeting of the Association for
Computational Linguistics: System Demonstrations, pages 31–36.
Association for Computational Linguistics.



Dyer, C. (2017).
Should neural network architecture reflect linguistic structure?
In Proceedings of the 21st Conference on Computational Natural
Language Learning (CoNLL 2017), page 1. Association for
Computational Linguistics.



Firth, J. (1957).
A synopsis of linguistic theory, 1930-1955.
Blackwell.

References III



Harris, Z. S. (1954).

Distributional structure.

Word, 10(2-3):146–162.



Le, Q. and Mikolov, T. (2014).

Distributed representations of sentences and documents.

In International Conference on Machine Learning, pages 1188–1196.



Levy, O. and Goldberg, Y. (2014).

Neural word embedding as implicit matrix factorization.




In Advances in neural information processing systems, pages 2177–2185.



Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013).

Distributed representations of words and phrases and their compositionality.

Advances in Neural Information Processing Systems 26.

-  Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1964).
The measurement of meaning.
University of Illinois Press.
-  Pennington, J., Socher, R., and Manning, C. D. (2014).
Glove: Global vectors for word representation.
In Empirical Methods in Natural Language Processing (EMNLP),
pages 1532–1543.
-  Rong, X. (2014).
word2vec parameter learning explained.
arXiv preprint arXiv:1411.2738.