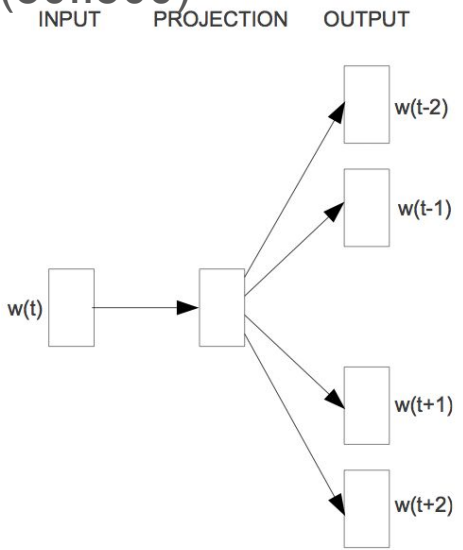
Усвоение множества смыслов в векторных представлениях слов

Embracing multiple senses in word space models

paul.khudan@youscan.io

Вступление. Универсальные векторные представления слов

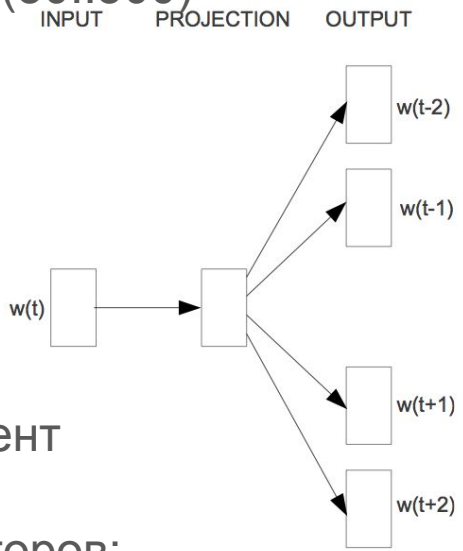
- Представление сущностей языка (слов, словосочетаний) в виде векторов действительных чисел низкой размерности (50..500)
- Единообразно выражают различные категории языка (морфология, синтаксис, семантика)
- Теория дистрибутивной семантики - смысл слова определяется его контекстами (CBOW, skip-gram)



Skip-gram

Вступление. Универсальные векторные представления слов

- Представление сущностей языка (слов, словосочетаний) в виде векторов действительных чисел низкой размерности (50..500)
- Единообразно выражают различные категории языка (морфология, синтаксис, семантика)
- Теория дистрибутивной семантики - смысл слова определяется его контекстами (CBOW, skip-gram)



Skip-gram

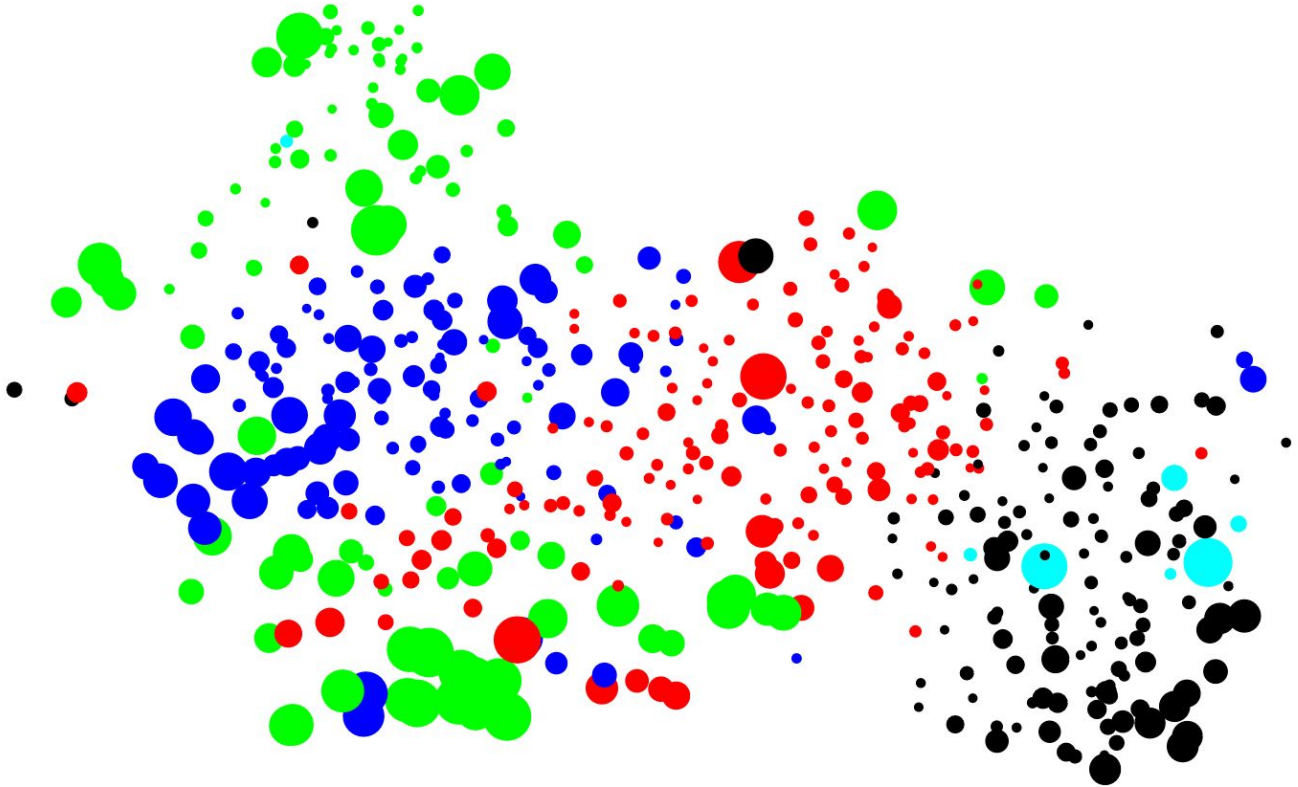
- Работа с векторами слов: синонимизация
- Анализ векторного пространства: выделение компонент смыслов, языковых категорий
- Построение сложных языковых моделей на базе векторов: классификаторы, DL …

Вступление. Векторные представления лишенные многозначности

- Усвоение множества значений слов (*Word Sense Induction*)
- Определение значения слова (*Word Sense Disambiguation*)
- Предоставление различных представлений (векторов) для разных значений слова
- *Без учителя и “сторонней” помощи

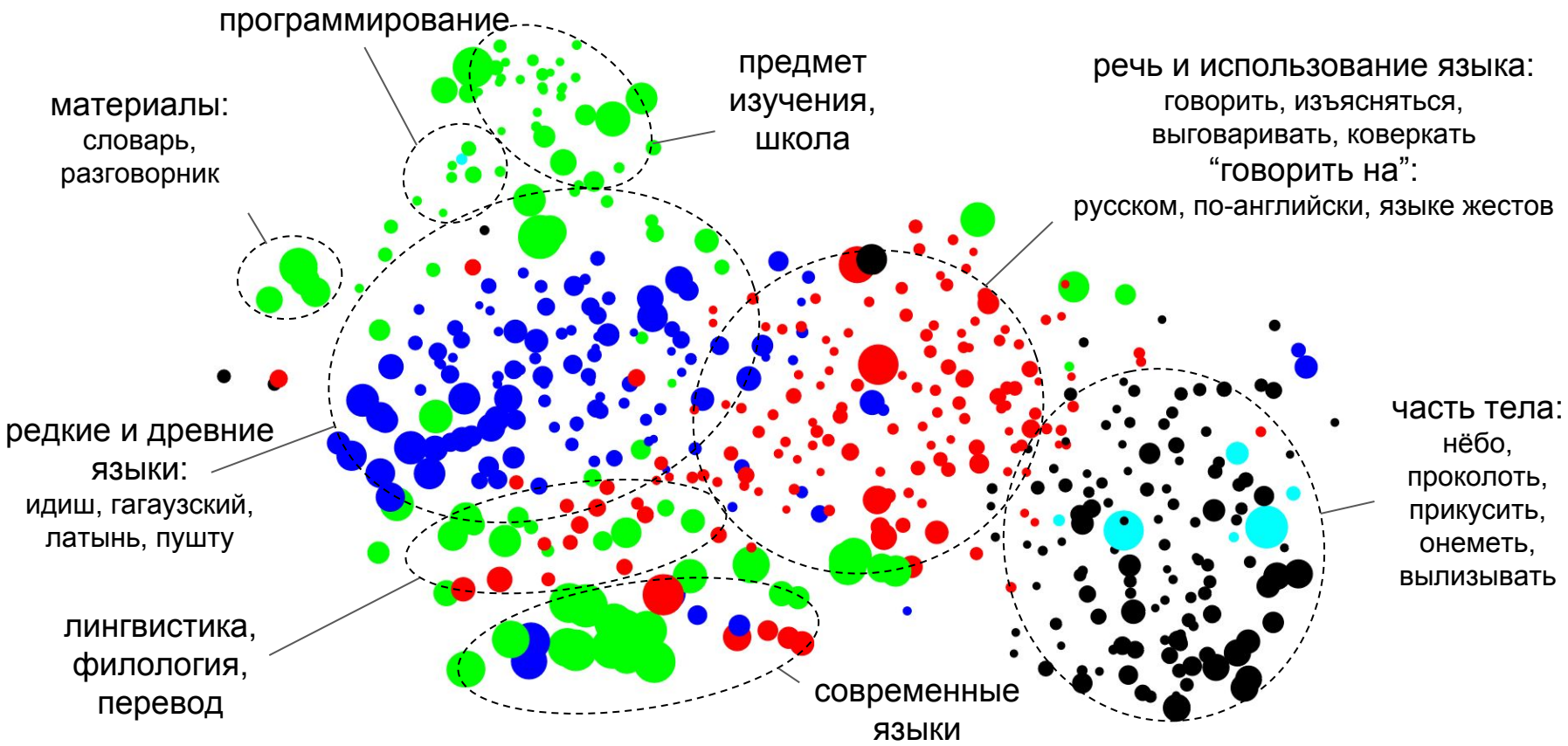
word2vec: skipgram, d300, w5, n5
cosine similarity, top500; rank-sized; cluster-colored; t-SNE

“язык”



word2vec: skipgram, d300, w5, n5
cosine similarity, top500; rank-sized; cluster-colored; t-SNE

“язык”



Проблемы I. Омонимия и многозначность

- Совпадение написания слов:
  - полные омонимы (*лук, среда*)
  - частичные омонимы (*к **трем** часам - **трем** на мелкой терке, **пила** двуручная - **пила** до приема пищи*)
  - омографы - слова, совпадающие по написанию, но различные по звучанию и значению: *кру́жки - кружки́, засы́пал - засыпа́л, па́рить - пари́ть*

Проблемы I. Омонимия и многозначность

- Совпадение написания слов:
  - полные омонимы (*лук, среда*)
  - частичные омонимы (*к **трем** часам - **трем** на мелкой терке, **пила** двуручная - **пила** до приема пищи*)
  - омографы - слова, совпадающие по написанию, но различные по звучанию и значению: *кру́жки - кружки́, засы́пал - засыпа́л, па́рить - пари́ть*

- Множество значений слова:
  - метафора - перенос наименования на основе сходства (формы, функции, атрибутов): *зеленый, мягкий, код, оценка, глушить*
  - метонимия - перенос на основе смежности (место-событие, действие-результат, форма-содержимое, …):
    Чернобыль, декларация, кухня

Проблемы I. Омонимия и многозначность

- Совпадение написания слов:
  - полные омонимы (*лук, среда*)
  - частичные омонимы (*к **трем** часам - **трем** на мелкой терке, **пила** двуручная - **пила** до приема пищи*)
  - омографы - слова, совпадающие по написанию, но различные по звучанию и значению: *кру́жки - кружки́, засы́пал - засыпа́л, па́рить - пари́ть*
- Множество значений слова:
  - метафора - перенос наименования на основе сходства (формы, функции, атрибутов): *зеленый, мягкий, код, оценка, глушить*
  - метонимия - перенос на основе смежности (место-событие, действие-результат, форма-содержимое, …): *Чернобыль, декларация, кухня*
- Набор “словарных” значений полисемичных слов
- Реализуемый смысл слова в высказывании; не зафиксирован словарями

Проблемы I. Омонимия и многозначность

Транснациональная корпорация
*построит зерновой терминал в **Украине**.*

*Второй мяч **Украина** забила в ворота сборной*
Израиля – 2:0.

***Украина** потребовала публично наказывать*
за нарушения Минска.

Подход I. Самостоятельное решение для слов

1. Разметка идиом / именованных сущностей / частей речи / значений слова через обучение с учителем и без учителя
2. Применение дополнительного фактора векторной модели как строго разделяющего (эксплицитно приписывается к словам до этапа построения векторной модели)

Подход I. Самостоятельное решение для слов

1. Разметка идиом / именованных сущностей / частей речи / значений слова через обучение с учителем и без учителя
2. Применение дополнительного фактора векторной модели как строго разделяющего (эксплицитно приписывается к словам до этапа построения векторной модели)

sense2vec (https://explosion.ai/blog/sense2vec-with-spacy)

Given: [‘New’, ‘York’] => ‘New York’ => new_york_GPE
In: most_similar(vec(new_york_GPE)):
Out: [(‘New York City’, 0.93), (‘Los Angeles’, 0.93), ..4.., (‘NY’, 0.89), ...]

https://demos.explosion.ai/sense2vec/?word=New%20York&sense=GPE

Подход I. Самостоятельное решение для слов

1. Разметка идиом / именованных сущностей / частей речи / значений слова через обучение с учителем и без учителя
2. Применение дополнительного фактора векторной модели как строго разделяющего (эксплицитно приписывается к словам до этапа построения векторной модели)

sense2vec (https://explosion.ai/blog/sense2vec-with-spacy)

Given: [‘New’, ‘York’] => ‘New York’ => new_york_GPE
In: most_similar(vec(new_york_GPE)):
Out: [(‘New York City’, 0.93), (‘Los Angeles’, 0.93), ..4.., (‘NY’, 0.89), ...]

https://demos.explosion.ai/sense2vec/?word=New%20York&sense=GPE

- Модель более точно описывает “смысловые” категории (лексико-иерархические связи)
- Зависимость от разметки при использовании модели

Проблемы II. Обратная эффективность векторной модели

- Векторные модели слов при обучении “захватывают” грамматические, синтаксические и семантические категории
- Есть возможность выделить соответствия между значениями векторов и упомянутыми категориями (пр.: слово-векторная арифметика)

Проблемы II. Обратная эффективность векторной модели

- Векторные модели слов при обучении “захватывают” грамматические, синтаксические и семантические категории
- Есть возможность выделить соответствия между значениями векторов и упомянутыми категориями (пр.: слово-векторная арифметика)

```
In: w2v.similar_by_vector(vec('москва') - vec('россия') + vec('украина'))
```

Проблемы II. Обратная эффективность векторной модели

- Векторные модели слов при обучении “захватывают” грамматические, синтаксические и семантические категории
- Есть возможность выделить соответствия между значениями векторов и упомянутыми категориями (пр.: слово-векторная арифметика)

```
In: w2v.similar_by_vector(vec('москва') - vec('россия') + vec('украина'))

Out: [('украина', 0.742128849029541),
      ('киев', 0.693192183971405),
      ('москва', 0.668310403823852 5),
      ('харьков', 0.620627105236053 5)]
```

Проблемы II. Обратная эффективность векторной модели

In: w2v.similar_by_vector(vec('она') - vec('он') + vec('мальчик'))

Проблемы II. Обратная эффективность векторной модели

```
In: w2v.similar_by_vector(vec('она') - vec('он') + vec('мальчик'))

Out: [('девочка', 0.8665761947631836),
      ('мальчик', 0.7871026396751404),
      ('мама', 0.7001113891601562),
      ('женщина', 0.69577026367187 5)]
```

Проблемы II. Обратная эффективность векторной модели

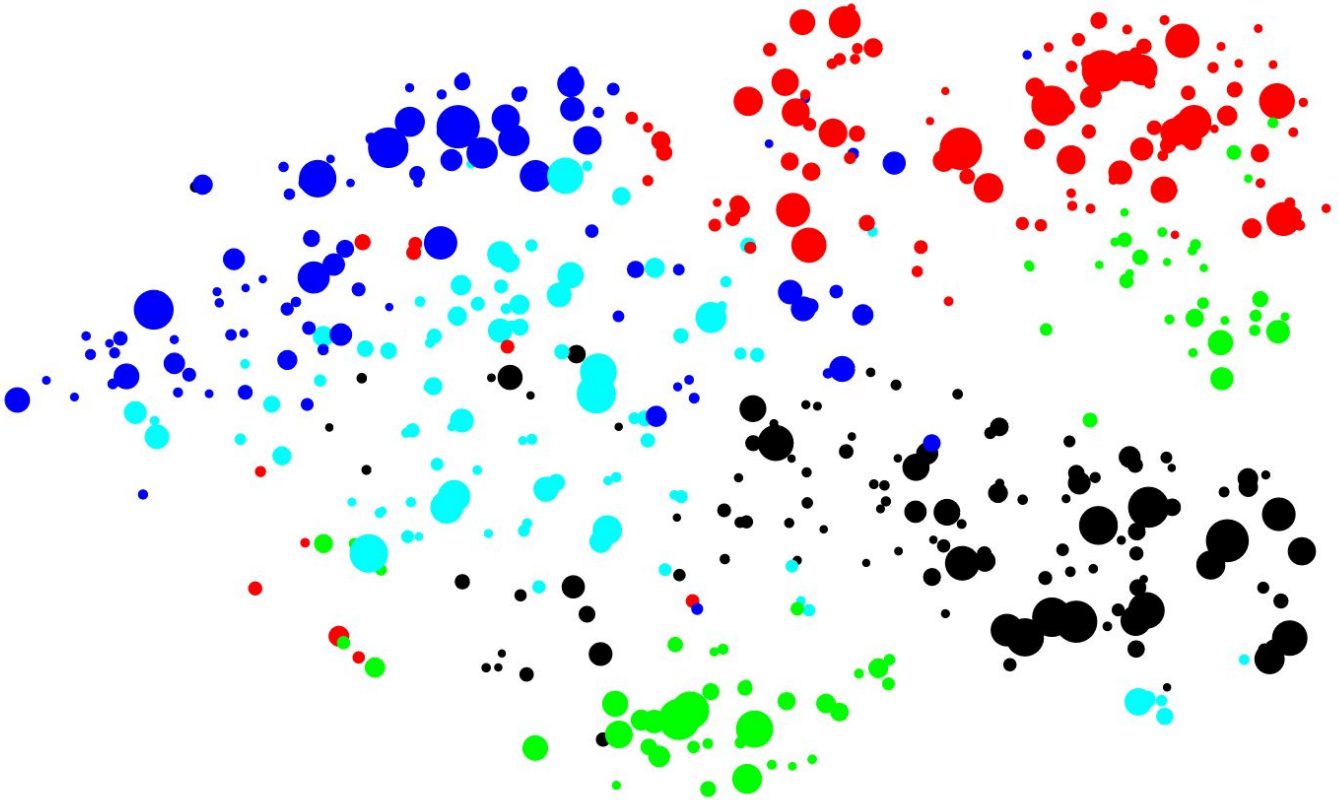
```
In: w2v.similar_by_vector(vec('она') - vec('он') + vec('мальчик'))

Out: [('девочка', 0.866576194763183 6),
      ('мальчик', 0.787102639675140 4),
      ('мама', 0.700111389160156 2),
      ('женщина', 0.695770263671875)]

In: w2v.similar_by_vector(vec('мальчик') - vec('девочка') + vec('она'))
```

Проблемы II. Обратная эффективность векторной модели

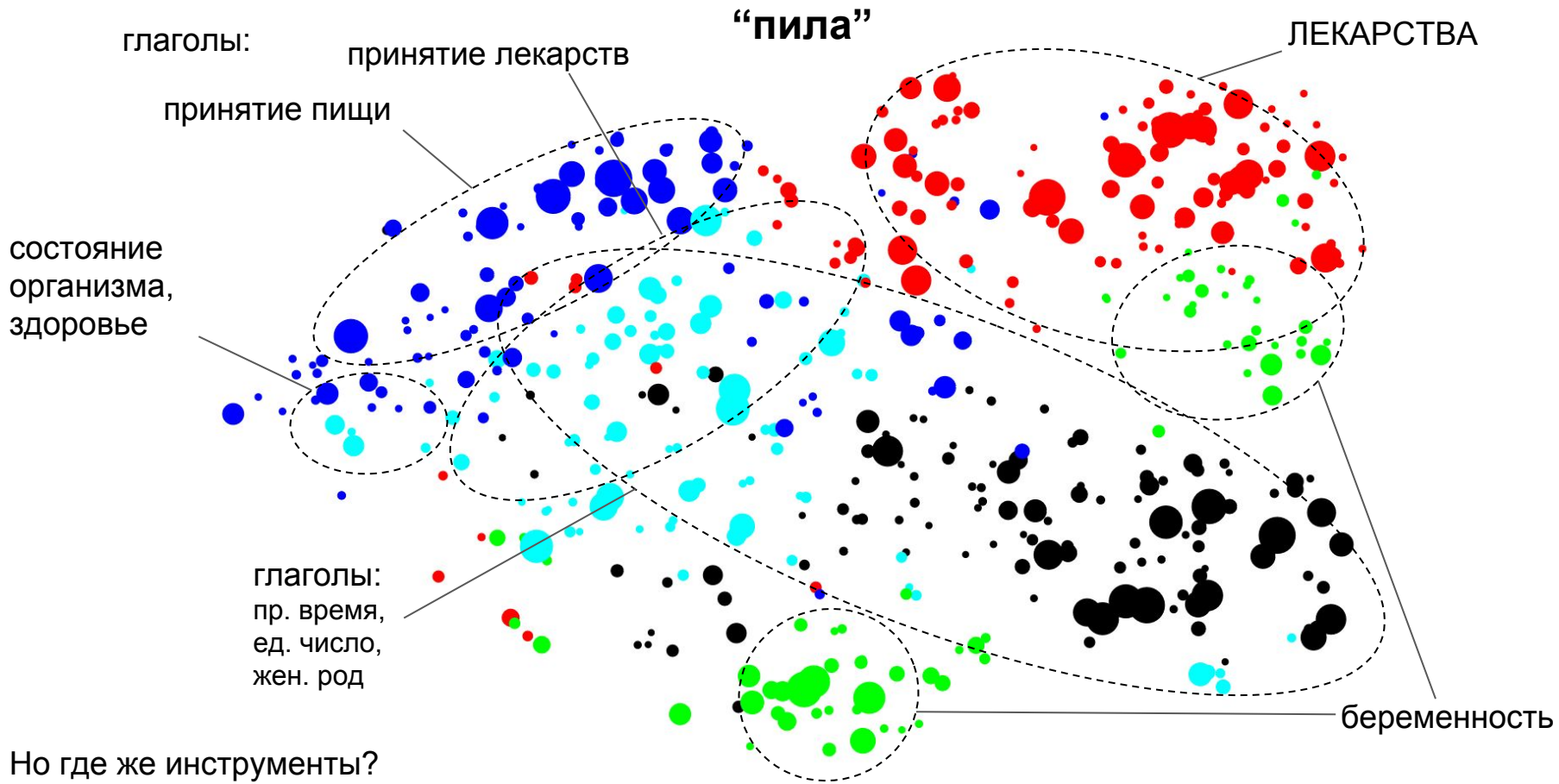
```
In: w2v.similar_by_vector(vec('она') - vec('он') + vec('мальчик'))

Out: [('девочка', 0.866576194763183 6),
      ('мальчик', 0.787102639675140 4),
      ('мама', 0.700111389160156 2),
      ('женщина', 0.695770263671875)]

In: w2v.similar_by_vector(vec('мальчик') - vec('девочка') + vec('она'))

Out: [('он', 0.868928432464599 6),
      ('отец', 0.749983191490173 3),
      ('парень', 0.734644174575805 7),
      ('кто-то', 0.726675152778625 5)]
```

word2vec: skipgram, d300, w5, n5; top 100k lexicon
cosine similarity, top500; rank-sized; cluster-colored; t-SNE

“пила”



word2vec: skipgram, d300, w5, n5; top 100k lexicon
cosine similarity, top500; rank-sized; cluster-colored; t-SNE



Но где же инструменты?

Проблемы II. Обратная эффективность векторной модели

- *“Unsupervised POS Induction with Word Embeddings”*, Chu-Cheng Lin, Waleed Ammar, Chris Dyer and Lori Levin, 2015
- *“Learning Hypernymy over Word Embeddings”*, Neha Nayak, 2015
- *“Learning Semantic Hierarchies via Word Embeddings”*, Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang and Ting Liu, 2014
- *“A Study of Neural Word Embeddings for Named Entity Recognition in Clinical Text”*, Yonghui Wu, Jun Xu, Min Jiang, Yaoyun Zhang, and Hua Xu, 2015

Проблемы II. Смещение домена - онлайн медиа и соц. сети

- Статистика словоупотребления изменена
  - Некоторые омонимы (см. “пила”) слабо представлены
  - Статистические модели других выборок уже не валидны

Проблемы II. Смещение домена - онлайн медиа и соц. сети

- Статистика словоупотребления изменена
  - Некоторые омонимы (см. “пила”) слабо представлены
  - Статистические модели других выборок уже не валидны
- Открытый словарь:
  - Новые сущности со старыми именами
    (*фуршет, Н. Тесла, корона*)
  - Новые имена старым сущностям
    (народные *“таз”*, *“мишка”*, *“смесь”*)

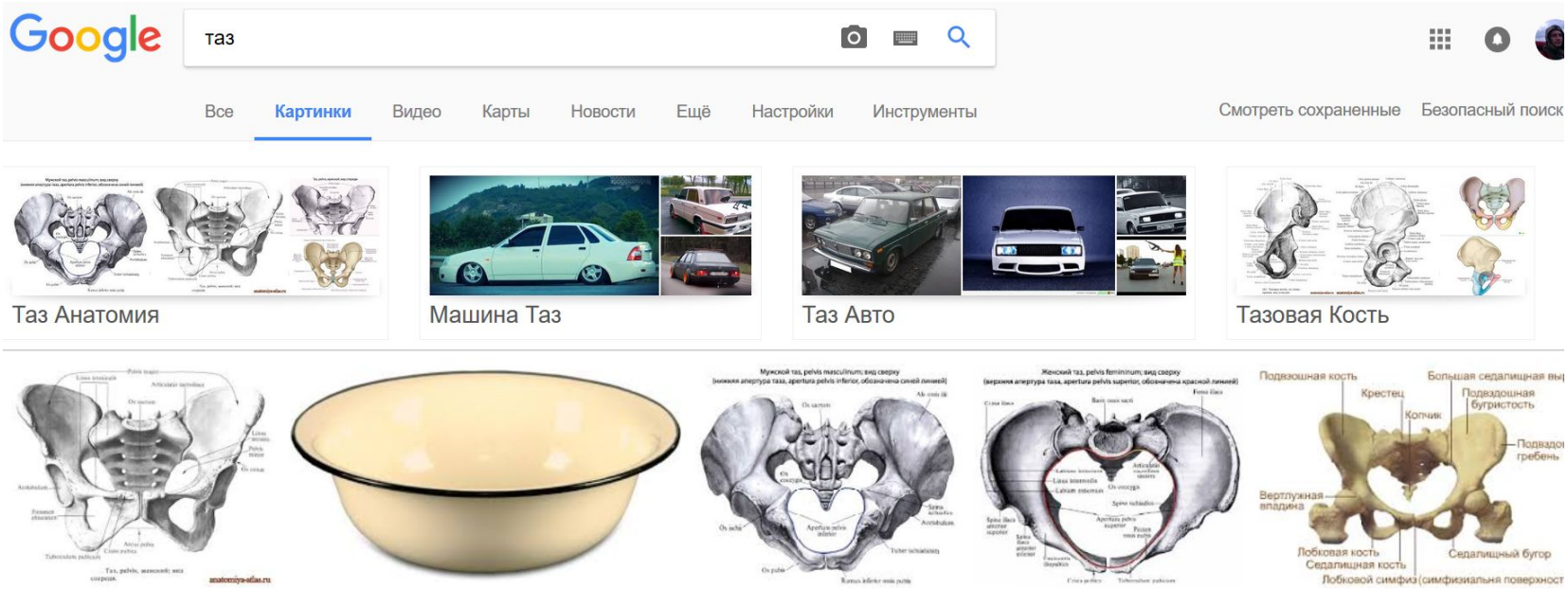
Проблемы II. Смещение домена - онлайн медиа и соц. сети

- Статистика словоупотребления изменена
  - Некоторые омонимы (см. “пила”) слабо представлены
  - Статистические модели других выборок уже не валидны
- Открытый словарь:
  - Новые сущности со старыми именами (*фуршет, Н. Тесла, корона*)
  - Новые имена старым сущностям (народные *“таз”*, *“мишка”*, *“смесь”*)
- Интернационализмы и межъязыковые омонимы: *гривня, шар, луна*

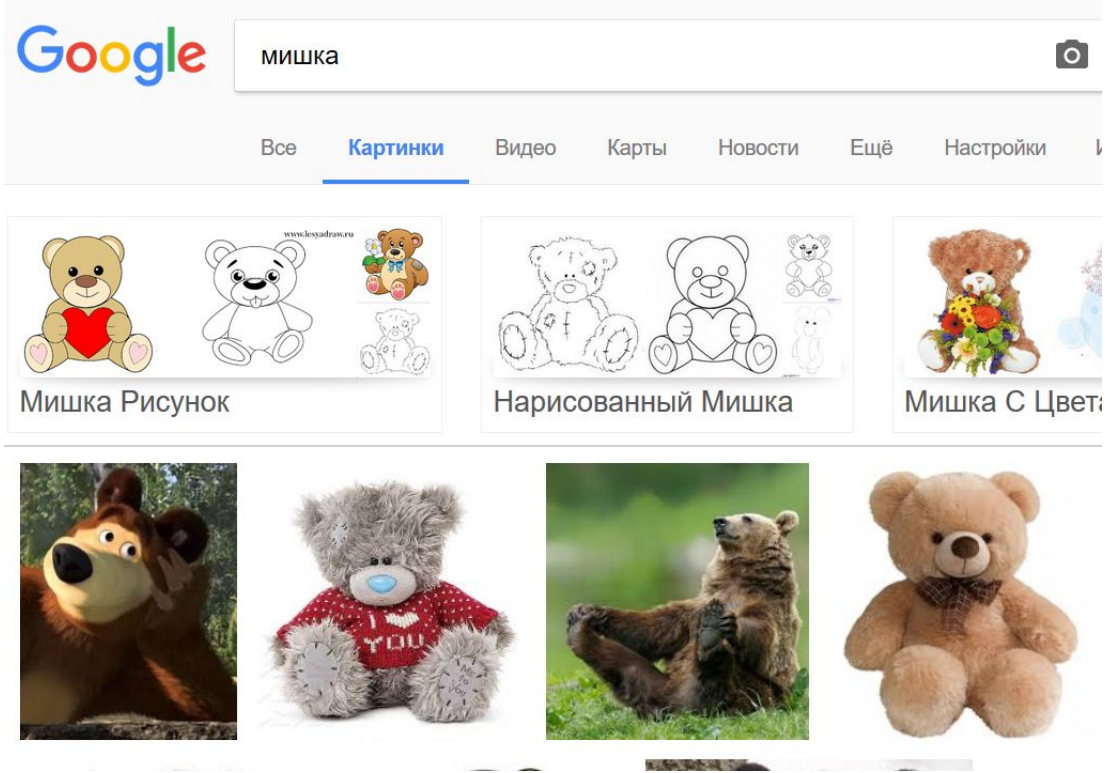
Проблемы II. Смещение домена - онлайн медиа и соц. сети

- Статистика словоупотребления изменена
  - Некоторые омонимы (см. “пила”) слабо представлены
  - Статистические модели других выборок уже не валидны
- Открытый словарь:
  - Новые сущности со старыми именами (*фуршет, Н. Тесла, корона*)
  - Новые имена старым сущностям (народные *“таз”*, *“мишка”*, *“смесь”*)
- Интернационализмы и межъязыковые омонимы: ***гривня, шар, луна***
- Орфография (прописные и строчные буквы, пунктуация, опечатки): ***едим-едем***, *...*

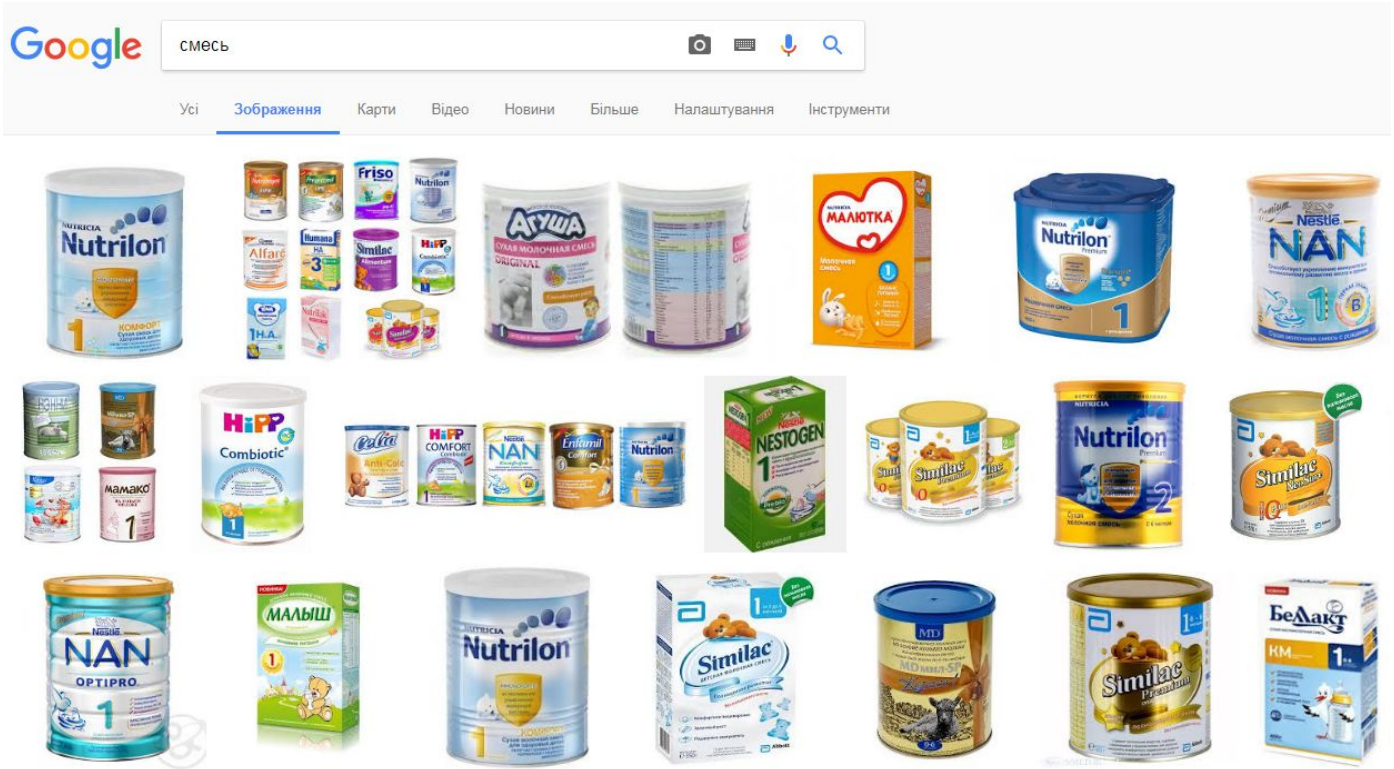
Проблемы II. Смещение домена - онлайн медиа и соц. сети



Проблемы II. Смещение домена - онлайн медиа и соц. сети

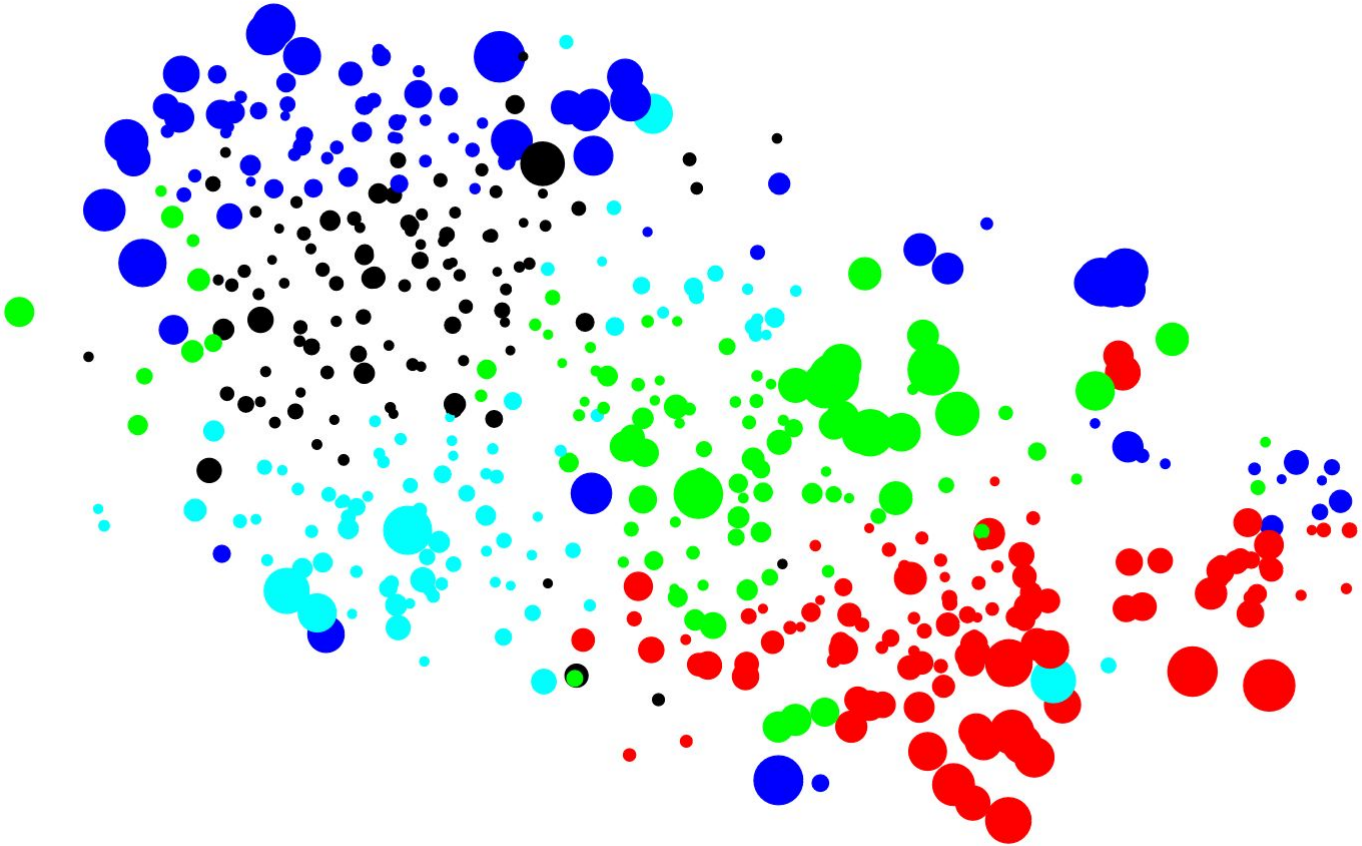


Проблемы II. Смещение домена - онлайн медиа и соц. сети



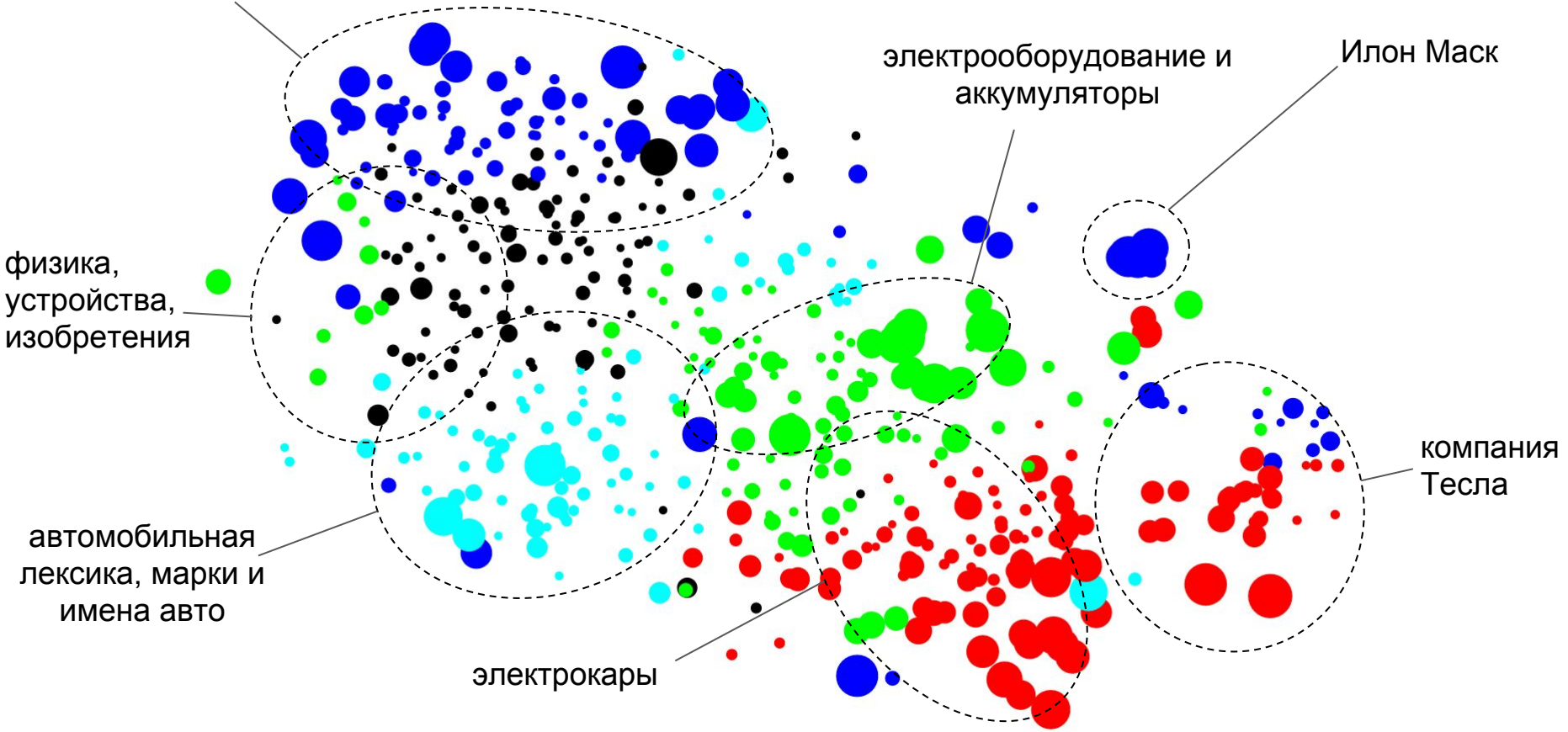
word2vec: skipgram, d300, w5, n5
cosine similarity, top500; rank-sized; cluster-colored; t-SNE

“тесла”



word2vec: skipgram, d300, w5, n5
cosine similarity, top500; rank-sized; cluster-colored; t-SNE

“тесла”



Подход II. Множество значений вектора слова. Интуитивно

1. Усвоение множества значений слова:
   → для каждого вектора слов найти *topK* похожих векторов
   → выделить кластеры отличных значений среди похожих
   → вывести вектора значений слова из векторов соотв. кластеров

Подход II. Множество значений вектора слова. Интуитивно

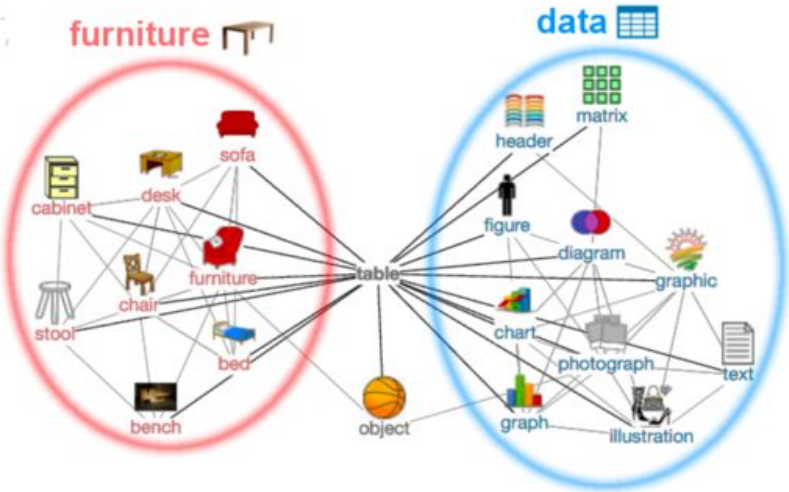
1. Усвоение множества значений слова:
   → для каждого вектора слов найти *topK* похожих векторов
   → выделить кластеры отличных значений среди похожих
   → вывести вектора значений слова из векторов соотв. кластеров
2. Предсказание реализованного смысла слова по контексту:
   → собрать вектора контекста слова
   → определить наиболее уместный
   вектор значения слова

Подход II. Множество значений вектора слова. Интуитивно

1. Усвоение множества значений слова:
   → для каждого вектора слов найти *topK* похожих векторов
   → выделить кластеры отличных значений среди похожих
   → вывести вектора значений слова из векторов соотв. кластеров

2. Предсказание реализованного смысла слова по контексту:
   → собрать вектора контекста слова
   → определить наиболее уместный вектор значения слова

SenseGram:



Подход II. SenseGram

- Формирования *эго-сети (размера N, связности n)* целевого слова-вектора *t* из *T* матрицы схожести всех векторов:
  - граф *G* из *V* узлов - *N* наиболее похожих векторов целевого вектора *t* из *T*
  - для каждого узла *v* из *V*:
    - граф *V'* размером *n* узлов - наиболее похожие вектора для *v* из *T*
    - для каждого узла *v'* из *V'*, добавить ребро *e(v, v')* в граф *G*, если *v'* принадлежит *V*

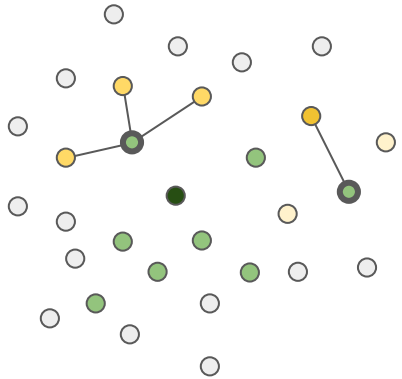
N=12, n=3

● ∈ *V*

○ ∉ *V*

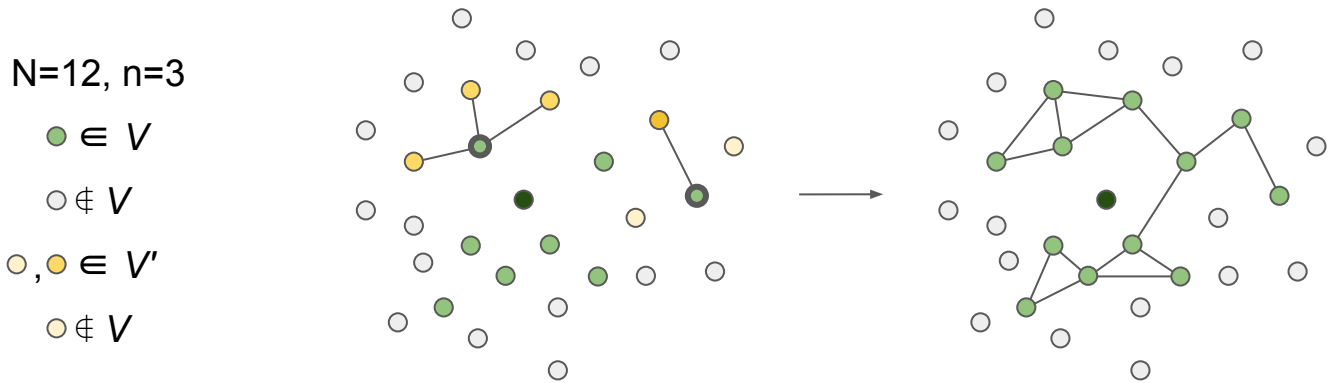
○,● ∈ *V'*

○ ∉ *V*



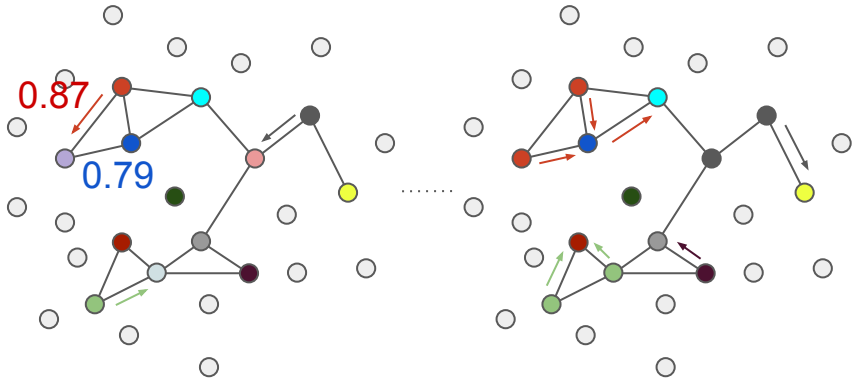
Подход II. SenseGram

- Формирования *эго-сети (размера N, связности n)* целевого слова-вектора *t* из *T* матрицы схожести всех векторов:
  - граф *G* из *V* узлов - *N* наиболее похожих векторов целевого вектора *t* из *T*
  - для каждого узла *v* из *V*:
    - граф *V'* размером *n* узлов - наиболее похожие вектора для *v* из *T*
    - для каждого узла *v'* из *V'*, добавить ребро *e(v, v')* в граф *G*, если *v'* принадлежит *V*



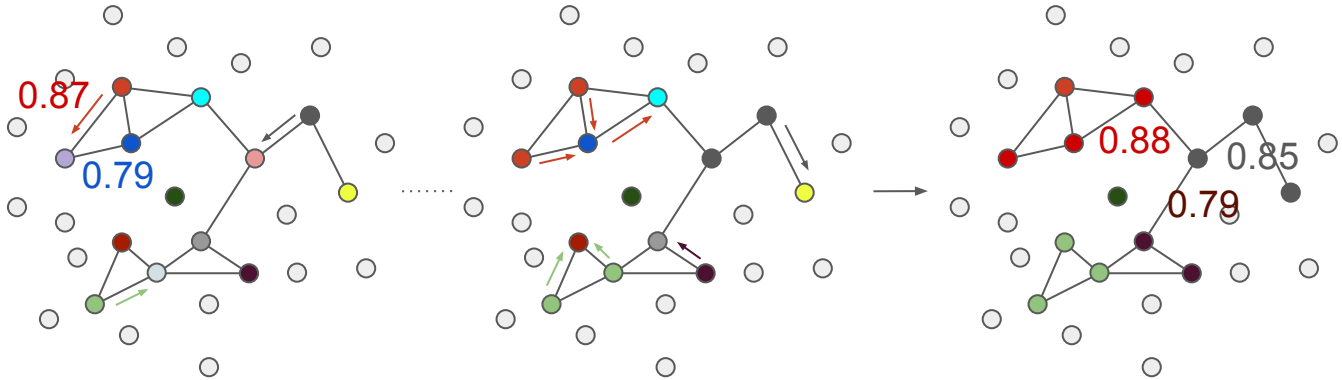
Подход II. SenseGram

- Кластеризация ненаправленного связного *графа G* методом *Chinese Whispers*:
  - всем узлам графа *G* присваивается случайный уникальный кластер c_v
  - в случайном порядке, для каждого узла *v* в *G*:
    - изменить c_v на такой c_v, узлы v_c' которого имеют наибольшее количество связей с вектором *v* (учитывать вес связей)
      - в случае равных v_c', выбрать случайный
  - повторять до схождения или *iter* раз



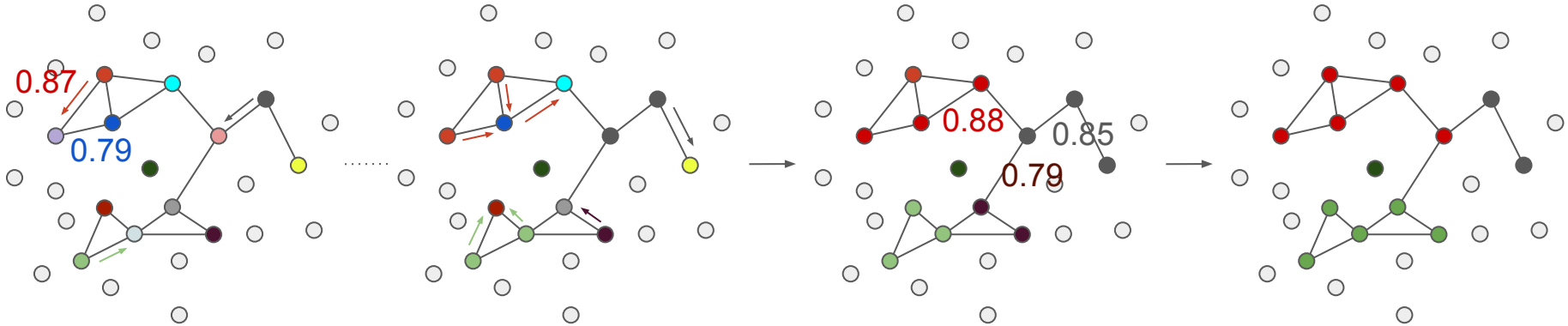
Подход II. SenseGram

- Кластеризация ненаправленного связного *графа G* методом *Chinese Whispers*:
  - всем узлам графа *G* присваивается случайный уникальный кластер c_v
  - в случайном порядке, для каждого узла *v* в *G*:
    - изменить c_v на такой $c_{v'}$, узлы v_c' которого имеют наибольшее количество связей с вектором *v* (учитывать вес связей)
      - в случае равных v_c', выбрать случайный
  - повторять до схождения или *iter* раз



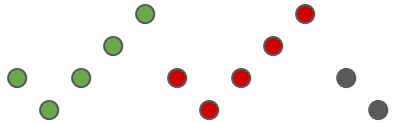
Подход II. SenseGram

- Кластеризация ненаправленного связного *графа G* методом *Chinese Whispers*:
  - всем узлам графа *G* присваивается случайный уникальный кластер c_v
  - в случайном порядке, для каждого узла *v* в *G*:
    - изменить c_v на такой c_v, узлы v_c' которого имеют наибольшее количество связей с вектором *v* (учитывать вес связей)
      - в случае равных v_c', выбрать случайный
  - повторять до схождения или *iter* раз



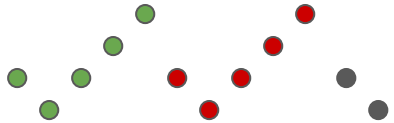
Подход II. SenseGram

- Отсеивание кластеров размером $< m$:



Подход II. SenseGram

- Отсеивание кластеров размером < *m*:



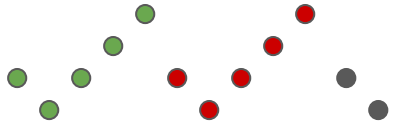
- Расчет векторов самостоятельных значений слова:

$$\mathbf{s}_i = \frac{\sum_{k=1}^{n} \gamma_i(w_k) vec_w(w_k)}{\sum_{k=1}^{n} \gamma_i(w_k)}$$

функция взвешивания

Подход II. SenseGram

- Отсеивание кластеров размером < *m*:



- Расчет векторов самостоятельных значений слова:

$$\mathbf{s}_i = \frac{\sum_{k=1}^{n} \gamma_i(w_k) vec_w(w_k)}{\sum_{k=1}^{n} \gamma_i(w_k)}$$

функция взвешивания

- Определение смысла слова по контексту:

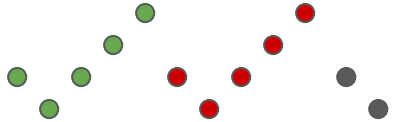
$$s^* = \arg\max_i P(C|\mathbf{s}_i) = \arg\max_i \frac{1}{1 + e^{-\bar{\mathbf{c}}_c \cdot \mathbf{s}_i}}$$

CBOW fashion

среднее векторов предсказываемого контекста, *syn1*

Подход II. SenseGram

- Отсеивание кластеров размером < *m*:



- Расчет векторов самостоятельных значений слова:

$$\mathbf{s}_i = \frac{\sum_{k=1}^{n} \gamma_i(w_k) vec_w(w_k)}{\sum_{k=1}^{n} \gamma_i(w_k)}$$

функция взвешивания

- Определение смысла слова по контексту:

$$s^* = \arg\max_i P(C|\mathbf{s}_i) = \arg\max_i \frac{1}{1 + e^{-\bar{\mathbf{c}}_c \cdot \mathbf{s}_i}}$$

CBOW fashion

среднее векторов предсказываемого контекста, *syn1*

$$s^* = \arg\max_i sim(\mathbf{s}_i, C) = \arg\max_i \frac{\bar{\mathbf{c}}_w \cdot \mathbf{s}_i}{\|\bar{\mathbf{c}}_w\| \cdot \|\mathbf{s}_i\|}$$

среднее векторов контекста

Подход II. SenseGram

- Построение эго-сети трудоемко (для каждого из слов)
- Чувствительность к параметрам
- Сложность кластеризации линейно растет

Подход II. SenseGram

- Построение эго-сети трудоемко (для каждого из слов)
- Чувствительность к параметрам
- Сложность кластеризации линейно растет

+ Имена/общеупотребительные слова: *Родни - родни*
+ Части речи: *одного - одного, господи - господи (междометие)*
+ Семантические категории: *наркотики, таз (2), близко, шар, замок*
+ Сокращения: *р(убль) - р(азмер)*
+ Слова из разных языков/интернационализмы: *гривня, тнг*

Подход II. SenseGram

- Построение эго-сети трудоемко (для каждого из слов)
- Чувствительность к параметрам
- Сложность кластеризации линейно растет

+ Имена/общеупотребительные слова: *Родни - родни*
+ Части речи: *одного - одного, господи - господи (междометие)*
+ Семантические категории: *наркотики, таз (2), близко, шар, замок*
+ Сокращения: *р(убль) - р(азмер)*
+ Слова из разных языков/интернационализмы: *гривня, тнг*

- Найдено в среднем 2 значения на слово
- Некоторые значения раздроблены (особенности стилистики?)
- Чувствительность к шуму

Подход II. SenseGram

- Implementation: https://github.com/tudarmstadt-lt/sensegram
- *“Making Sense of Word Embeddings”*, http://aclweb.org/anthology/W/W16/W16-1620.pdf
- *“Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems”*, https://pdfs.semanticscholar.org/3e71/0251cb01ba6e1c0c735591776a212edc461f.pdf

Подход III. Adaptive Skip-gram

- Байесовская модификация модели skip-gram со скрытой переменной ***z***, $Z = \{z_i\}_{i=1}^{N}$, индексом значения употребленного слова, т.е. смыслом
- Векторные представления имеют вид $\theta = \{In(x_i, z_i), Out(x_i)\}$
- Модель асимметрична - $Out(x_i)$ являются векторами слов, без знаний о значениях слов из $In(x_i, z_i)$

$$p(Y, Z, \beta | X, \alpha, \theta) = \prod_{w=1}^{V} \prod_{k=1}^{\infty} p(\beta_{wk}|\alpha) \prod_{i=1}^{N} \left[p(z_i|x_i, \beta)) \prod_{j=1}^{C} p(y_{ij}|z_i, x_i, \theta) \right]$$

Подход III. Adaptive Skip-gram

- Байесовская модификация модели skip-gram со скрытой переменной ***z***, $Z = \{z_i\}_{i=1}^{N}$, индексом значения употребленного слова, т.е. смыслом
- Векторные представления имеют вид $\theta = \{In(x_i, z_i), Out(x_i)\}$
- Модель асимметрична - $Out(x_i)$ являются векторами слов, без знаний о значениях слов из $In(x_i, z_i)$

$$p(Y, Z, \beta | X, \alpha, \theta) = \prod_{w=1}^{V} \prod_{k=1}^{\infty} p(\beta_{wk}|\alpha) \prod_{i=1}^{N} \left[p(z_i|x_i, \beta)) \prod_{j=1}^{C} p(y_{ij}|z_i, x_i, \theta) \right]$$

Гиперпараметр, коэффициент плотности распределений в процессе Дирихле

Оптимизируемый параметр - векторные представления

Априорная вероятность *k*-го значения *слова* ***w***, задается *stick-breaking* процессом Дирихле

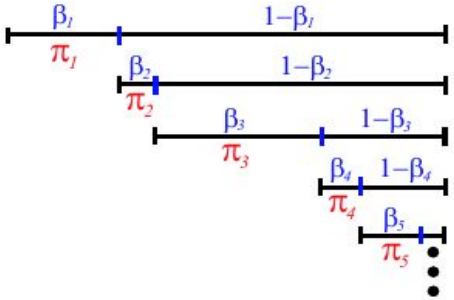
Вероятность наблюдения x_i *быть употребленным в смысле* z_i

Вероятность контекста y_{ij} у наблюдения x_i со смыслом z_i, где ***C*** - ширина контекста; skip-gram часть

Подход III. Adaptive Skip-gram

- Априорные вероятности значений слов задаются непрерывным многомерным процессом Дирихле - stick-breaking Dirichlet process:

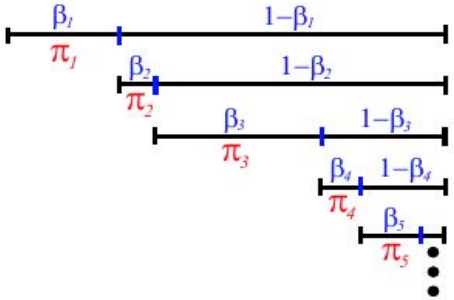
$$p(z = k|w, \boldsymbol{\beta}) = \beta_{wk} \prod_{r=1}^{k-1} (1 - \beta_{wr}), \quad p(\beta_{wk}|\alpha) = \text{Beta}(\beta_{wk}|1, \alpha), \quad k = 1, \ldots$$



Подход III. Adaptive Skip-gram

- Априорные вероятности значений слов задаются непрерывным многомерным процессом Дирихле - stick-breaking Dirichlet process:

$$p(z = k|w, \beta) = \beta_{wk} \prod_{r=1}^{k-1} (1 - \beta_{wr}), \quad p(\beta_{wk}|\alpha) = \text{Beta}(\beta_{wk}|1, \alpha), \quad k = 1, \ldots$$



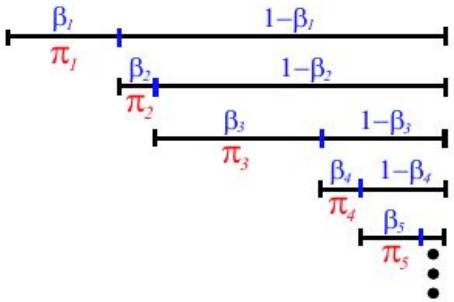
Chinese restaurant process:
- rich get richer:
  $p(z = k) = n_k / (n_z + \alpha)$
- новый кластер всегда возможен:
  $p(z = k+1) = \alpha / (n_z + \alpha)$

- На практике, *k* ограничивается T теоретическим максимальным кол-вом значений у слов. Стартовое кол-во значений для всех слов - 1
- Меньшие значения *α* порождают более плотные распределения и как следствие - меньшее дробление на новые значения слов

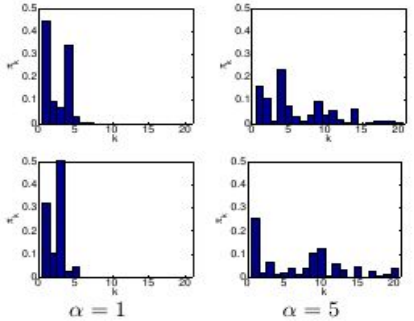
Подход III. Adaptive Skip-gram

- Априорные вероятности значений слов задаются непрерывным многомерным процессом Дирихле - stick-breaking Dirichlet process:

$$p(z = k|w, \beta) = \beta_{wk} \prod_{r=1}^{k-1} (1 - \beta_{wr}), \quad p(\beta_{wk}|\alpha) = \text{Beta}(\beta_{wk}|1, \alpha), \quad k = 1, \ldots$$



Chinese restaurant process:
- rich get richer:
  $p(z = k) = n_k / (n_z + \alpha)$
- новый кластер всегда возможен:
  $p(z = k+1) = \alpha / (n_z + \alpha)$



- На практике, *k* ограничивается T теоретическим максимальным кол-вом значений у слов. Стартовое кол-во значений для всех слов - 1
- Меньшие значения *α* порождают более плотные распределения и как следствие - меньшее дробление на новые значения слов

Подход III. Adaptive Skip-gram

- Предсказание реализованного смысла слова по контексту:

$$p(z = k|x, \mathbf{y}, \theta) \propto p(\mathbf{y}|x, k, \theta) \int p(k|\boldsymbol{\beta}, x) q(\boldsymbol{\beta}) d\boldsymbol{\beta}$$

Подход III. Adaptive Skip-gram

- Предсказание реализованного смысла слова по контексту:

$$p(z = k|x, \mathbf{y}, \theta) \propto p(\mathbf{y}|x, k, \theta) \int p(k|\boldsymbol{\beta}, x) q(\boldsymbol{\beta}) d\boldsymbol{\beta}$$

- Предсказание контекста по слову и индексу значения:

$$p(\mathbf{y}|x, \mathcal{D}, \theta, \alpha) = \int \sum_{z=1}^{T} p(\mathbf{y}|x, z, \theta) p(z|\boldsymbol{\beta}, x) q(\boldsymbol{\beta}) d\boldsymbol{\beta}$$

Подход III. Adaptive Skip-gram

- Скорость обучения в T раз ниже чем у skip-gram
- В среднем - 3.8 значения на слово, с $\alpha=0.1$
- При $T=5$, некоторые слова имели все 5 вероятных значений
- Чувствительность к шуму

Подход III. Adaptive Skip-gram

- Скорость обучения в *T* раз ниже чем у skip-gram
- В среднем - 3.8 значения на слово, с *α=0.1*
- При *T=5*, некоторые слова имели все 5 вероятных значений
- Чувствительность к шуму
- *Замок* - 5 значений:
  - подвал, люк, потайной, вестибюль, тоннель (шум?)
  - сигнализация, автозапуском, противотуманки, центральный, автозапуск
  - лес, панда, герой, капитан (шум)
  - съёмный, двойной, ремешок, молния, сьемный, ручка, потайной, накладной
  - слоуп, прага, вары, дворец, мраморный, кукс, литомышль, пльзень, кутна

Подход III. Adaptive Skip-gram

- Julia implementation: https://github.com/sbos/AdaGram.jl (hotfix @ https://github.com/buriy/AdaGram.jl/)
- Python implementation (raw): https://github.com/lopuhin/python-adagram
- *“Breaking Sticks and Ambiguities with Adaptive Skip-Gram”*, https://arxiv.org/pdf/1502.07257.pdf
- https://www.slideshare.net/DenisDus1/word2vec-2
- http://www.skoltech.ru/app/data/uploads/sites/2/2016/05/VetrovDLWorkshop.pdf
- Demo from authors, russian language: http://adagram.ll-cl.org/

Appendix

- *“Do Multi-Sense Embeddings Improve Natural Language Understanding?”*, https://nlp.stanford.edu/pubs/emnlp2015_1_jiwei.pdf
- 2017, word2gm, *“Multimodal Word Distributions”*, http://www.aclweb.org/anthology/P/P17/P17-1151.pdf, github inside
- 2017, *“A Simple Approach to Learn Polysemous Word Embeddings”,* https://arxiv.org/pdf/1707.01793v2.pdf, github inside

Thank You