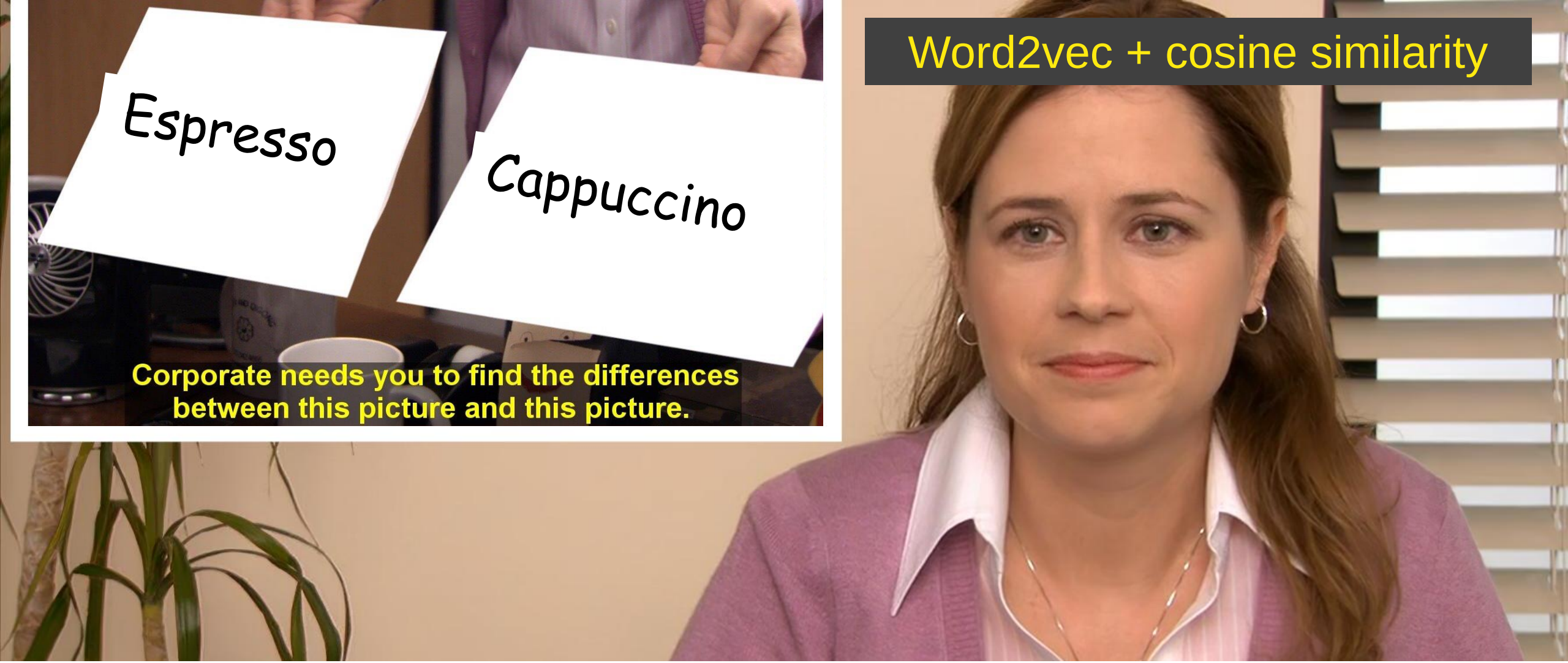


Приложения векторных представлений слов.

Попов Артём, OzonMasters, весна 2022

Natural Language Processing

Напоминание: векторные представления слов

Векторным представлением (эмбеддингом) слова $w \in W$ называется вектор $v_w \in \mathbb{R}^m$, где W – словарь коллекции, а m – размер представления.

На прошлом занятии мы изучили несколько подходов к построению эмбеддингов:

- SVD, Glove
- CBOW, Skip-gram
- FastText

Интерпретация векторных представлений

Что такое Intrinsic и Extrinsic подходы?

Skip-gram как count-based метод

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{c \in C(i)} \log p(c|w_i) = \sum_{w \in W} \sum_{c \in W} n_{wc} \log p(c|w) =$$

$$= \sum_{w \in W} n_w \sum_{c \in W} \frac{n_{wc}}{n_w} \log p(c|w) \to \max_{V,U}$$

Skip-gram как count-based метод

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{c \in C(i)} \log p(c|w_i) = \sum_{w \in W} \sum_{c \in W} n_{wc} \log p(c|w) =$$

$$= \sum_{w \in W} n_w \sum_{c \in W} \frac{n_{wc}}{n_w} \log p(c|w) \to \max_{V,U}$$

Добавление константы не меняет оптимизационную задачу:

$$\sum_{w \in W} n_w \sum_{c \in W} \frac{n_{wc}}{n_w} \left(\log p(c|w) - \log \frac{n_{wc}}{n_w} \right) =$$

$$= - \sum_{w \in W} n_w \sum_{c \in W} \hat{p}(c|w) \left(\log \frac{\hat{p}(c|w)}{p(c|w)} \right) \to \max_{V,U}$$

Skip-gram как count-based метод

Запишем функционал как минимизацию взвешенной KL-дивергенции:

$$\sum_{w \in W} n_w \sum_{c \in W} \frac{n_{wc}}{n_w} \left(\log \frac{\hat{p}(c|w)}{p(c|w)} \right) = \sum_{w \in W} n_w \sum_{c \in W} KL(\hat{p}(c|w) || p(c|w)) \to \min_{V,U}$$

Skip-gram это матричное разложение матрицы $X_{cw} = \hat{p}(c|w)$

Интересный факт. Тематическая модель PLSA имеет схожий функционал при специальном задании коллекции.

Интерпретация негативного сэмплирования

Напоминание. Функционал skip-gram negative sampling:

$$\sum_{i=1}^{N}\left(\sum_{c\in C(i)}\log p(1|c,w_i)+\sum_{c'\sim p(w)^{3/4}}\log p(0|c',w_i)\right)\to\max_{\text{U,V}}$$

Утверждение (Леви).

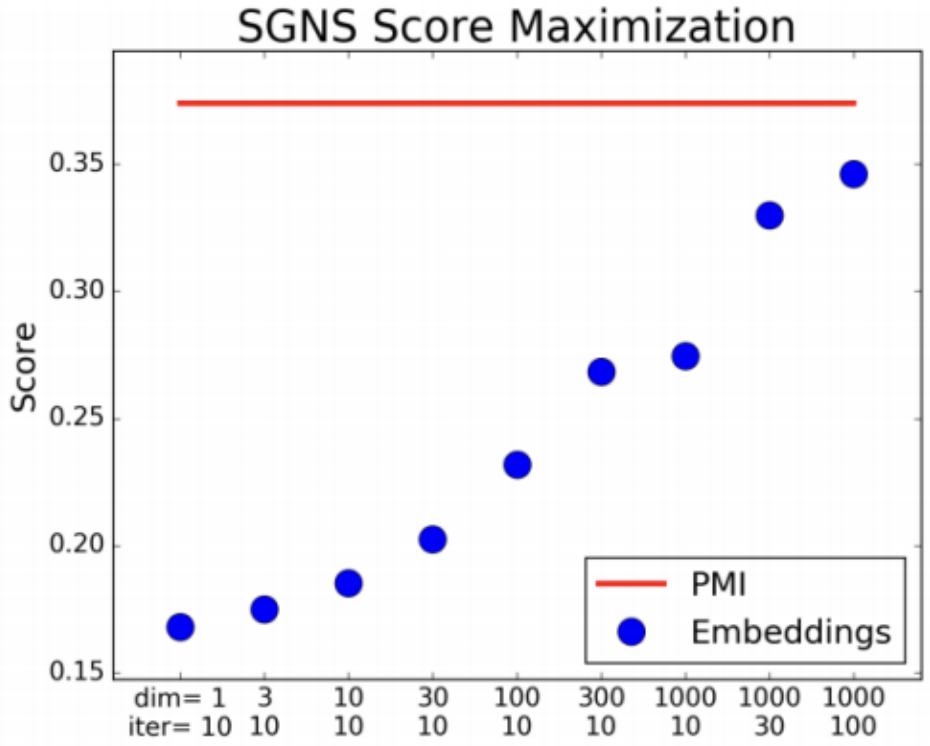
Пусть для любых $w, c \in W$ результат $\langle v_w, u_c\rangle$ не зависит от других пар слов. Тогда, в точке максимума SGNS для любых $w, c \in W$ будет выполнено:

$$\langle v_w, u_c\rangle = PMI(w,c) - \log k$$

Levy et al (NIPS 2014), Neural Word Embedding as Implicit Matrix Factorization

Интерпретация skip-gram negative sampling

На практике эффект
наблюдается
при больших размерностях.



Melamud et al (ACL 2017), Information-Theory Interpretation of the Skip-Gram Negative-Sampling Objective Function

Оценивание качества векторных представлений слов

Что такое Intrinsic и Extrinsic подходы?

Extrinsic оценивание представлений

Фиксируем: постановку задачи, данные для обучения и тестирования и архитектуру для решения задачи.

Подставляем в архитектуру разные типы эмбеддингов и **сравниваем** качество.

Пример. Используем задачу классификации с метрикой accuracy на датасете 20newsgroups. В качестве архитектуры используем линейный классификатор поверх усреднённых эмбеддингов слов.

Intrinsic оценивание представлений: близость

Оцениванием качества представлений на задачах, которые не требуют наличия дополнительной архитектуры.

Задача близости

Данные. Список из пар слов w, u и близостью между ними, посчитанной асессорами.

Модель. Измеряем близость между парами слов, например $\cos(v_w, v_u)$ или $\langle v_w, v_u \rangle$

Метрика. Считаем корреляцию Спирмена между списками близости согласно модели и согласно асессорам.

Примеры. Датасет близости wordsim353

первое слово	второе слово	близость
book	paper	7.46
five	month	3.38
king	cabbage	0.23
king	queen	8.58
money	dollar	8.42
cup	article	2.40
computer	laboratory	6.78

Intrinsic оценивание представлений: аналогии

Задача аналогий

Данные. Список четвёрок слов w_1, w_2, w_3, w_4, в котором w_1 относится к w_2 так же, как и w_3 к w_4

Модель. Находим самое близкое слово к $v(w_3) - v(w_1) + v(w_2)$ кроме самих слов w_1, w_2, w_3

Метрика. Доля правильно найденных слов

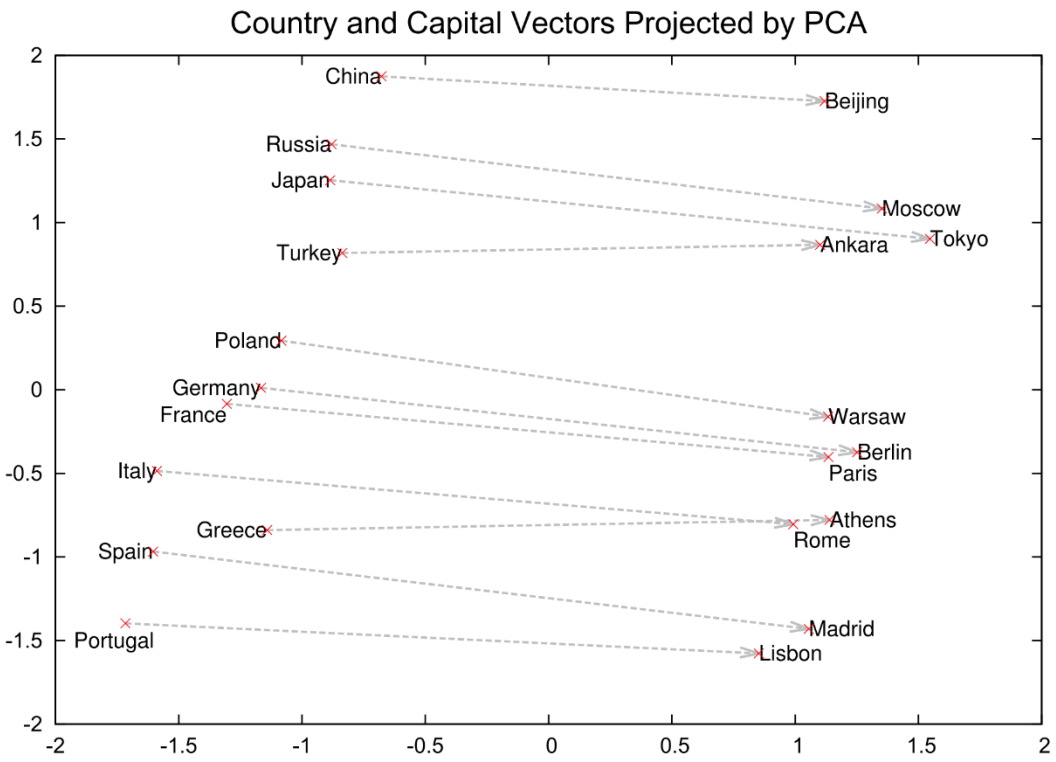
Семантика: $v(king) - v(boy) + v(girl) \approx v(queen)$

Синтаксис: $v(kings) - v(king) + v(queen) \approx v(queens)$

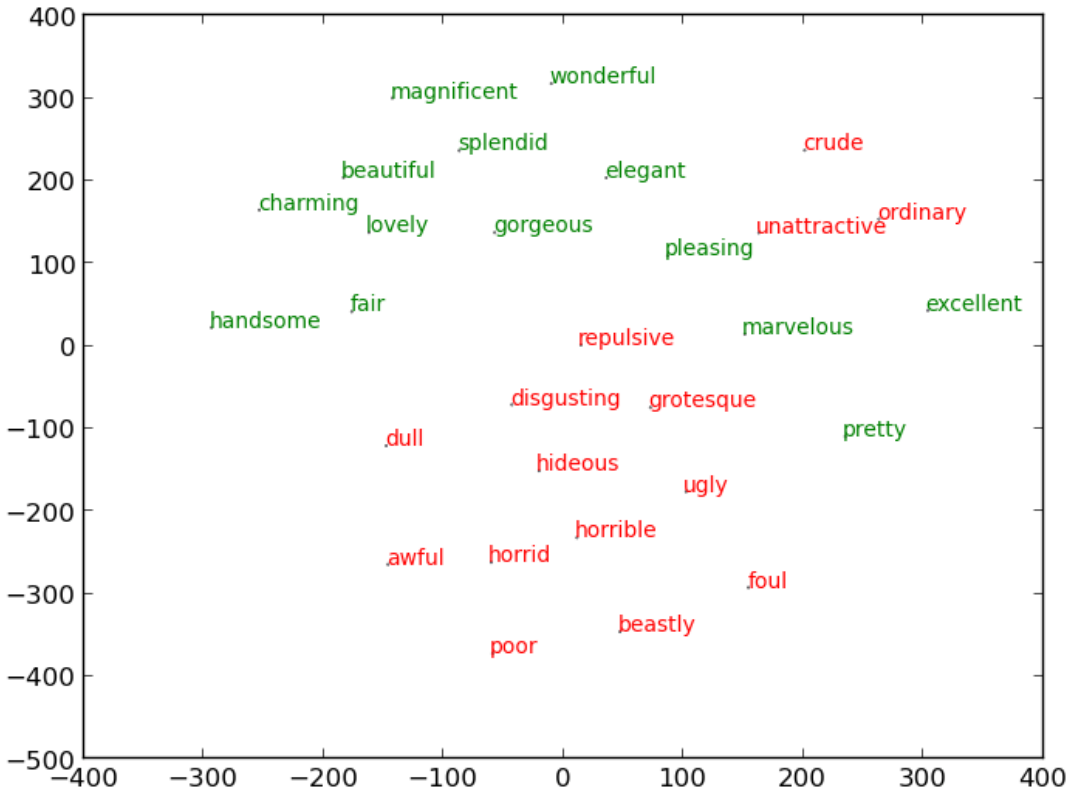
Linzen (2016), Issues in evaluating semantic spaces using word analogies

Rogers et. al. (*SEM 2017), The (Too Many) Problems of Analogical Reasoning with Word Vectors

Как можно понимать задачу аналогий?



PCA



TSNE

Так как же оценивать представления?

- Качество на intrinsic задачах слабо коррелирует с качеством решения итоговой задачи
- Intrinsic подход может быть полезен для быстрой оценки модели (проверить, что слова, специфичные для вашего датасета, имеет адекватных ближайших соседей)
- Intrinsic подход может быть полезен при интерпретации ошибок
- При extrinsic подходе, необходимо учитывать влияние архитектуры модели, решающей задачу
- В качестве extrinsic задачи лучше использовать конечную задачу, для которой строится решение

Эксперимент

Обучение на разных коллекциях

Эксперимент

Рассмотрим модели, обученные по двум датасетам:

- статьи Википедии + Национальный корпус русского языка
- статьи сайта Lurkmore (3.5К статей)

Для Википедии используем модель с сайта RusVectores.

Для Lurkmore обучим модель с нуля с помощью пакета Gensim.

Идея позаимствована из лекции Мурата Апишева для курса «Анализ Неструктурированных данных» ФКН ВШЭ.

Детали предобработки

Коллекция Луркморье:

- Все символы кроме букв были удалены
- Все слова лемматизированы (pymorphy2)
- Один документ — один абзац (важно при учёте контекста)
- Абзацы меньше двух слов были удалены

Коллекция Википедии:

- Все слова лемматизированы (UDPipe)
- Каждое слово преобразовано в слово_{часть речи}

Похожие слова

most_similar(россия_PROPN)

страна 0.695

европа 0.679

российский 0.604

франция 0.582

германия 0.574

most_similar(россия)

ссср 0.759

сша 0.754

германия 0.741

рашка 0.730

грузия 0.719

most_similar(полковник_NOUN)

подполковник 0.904

майор 0.875

генерал 0.805

генерал-майор 0.799

ротмистр 0.770

most_similar(полковник)

генерал 0.648

подполковник 0.647

майор 0.599

генералмайор 0.573

адмирал 0.557

Похожие слова

most_similar(тролль_NOUN)

гном 0.661

троллый 0.656

эльф 0.627

тролли 0.609

гоблин 0.589

most_similar(тролль)

троллинг 0.668

лурко** 0.538

провокатор 0.530

фрик 0.517

быдло 0.516

most_similar(музыка_NOUN)

мелодия 0.702

джаз 0.669

пение 0.649

песня 0.642

танец 0.630

most_similar(музыка)

мелодия 0.668

рэп 0.647

попёс 0.642

песнь 0.641

звук 0.630

Похожие слова

most_similar(мгу_PROPN)

мгу 0.843

лгу 0.773

м::в::ломоносов 0.728

мпгу 0.701

спбгу 0.697

most_similar(мгу)

университет 0.755

вуз 0.665

пту 0.656

мгимо 0.646

аспирант 0.640

most_similar(физтех_PROPN)

физтех_NOUN 0.701

мфти 0.694

мифи 0.632

физтех_DET 0.580

мирэа 0.578

most_similar(физтех)

мехмат 0.537

мифь 0.524

мгимо 0.518

мгу 0.502

филфак 0.496

Арифметические операции (триплеты)

яндекс - россия + сша

гугл 0.518

yahoo 0.467

пентагон 0.464

symantec 0.443

яндексяча 0.441

гугл 0.593

google 0.508

гуголь 0.504

rm 0.502

кэш 0.497

король - мужчина + женщина

королева_NOUN 0.754

королева_ADV 0.672

принц 0.627

королева_ADJ 0.625

король 0.623

император 0.583

королевский 0.555

фараон 0.548

халиф 0.523

герцог 0.523

Приложения

Поиск близких документов и классификация

Как можно использовать эмбеддинги?

1. Решение задачи поиска близких слов
2. Построение векторного представления документа
   Внимание. Векторным представлением документа может быть и вектор, и матрица (последовательность векторов)
3. Использование представлений в сложной архитектуре
4. Использование представлений для инициализации части весов в сложной архитектуре

Векторное представление документа

Самый простой и очевидный вариант – усреднение (сумма):

$$v_d = \frac{1}{|d|} \sum_{w \in d} v_w$$

Для разного учёта редких и частных слов, можно пробовать взвешенное усреднение (сумму):

$$v_d = \frac{1}{\sum_{w \in d} \alpha_{wd}} \sum_{w \in d} \alpha_{wd} v_w, \qquad \alpha_w \geq 0$$

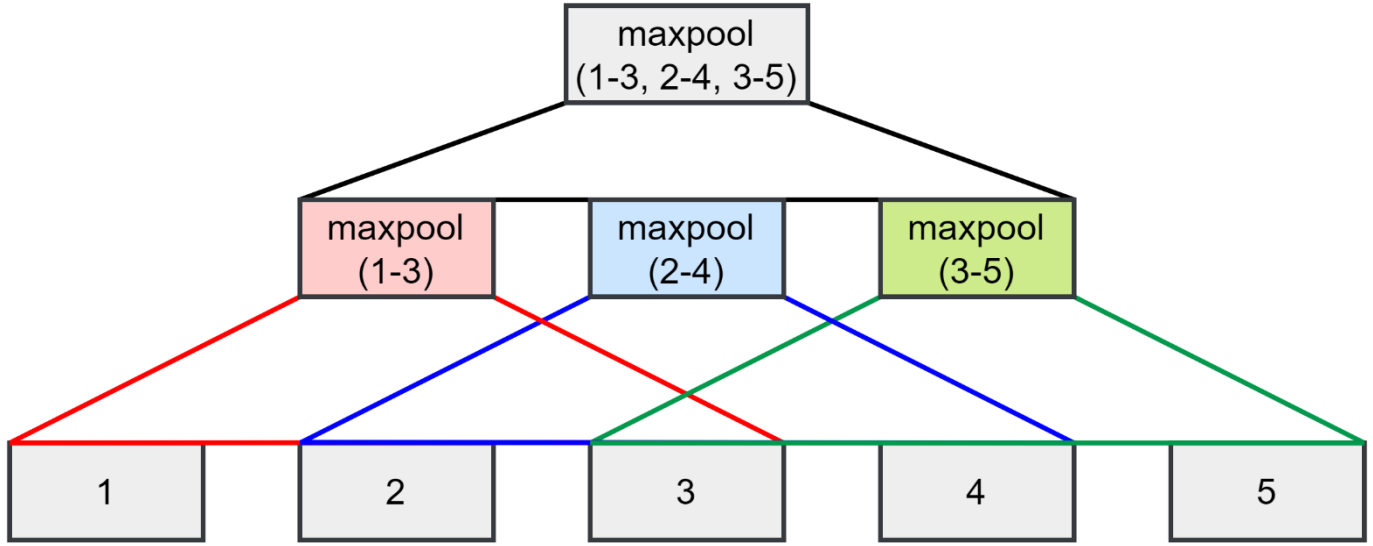
В качестве α_{wd} часто используют iDF значения.

Векторное представление документа

Можно использовать max-pooling / min-pooling / их конкатенацию:

$$v_{d,j} = \max\left(\left\{v_{w,j} \,\middle|\, w \in d\right\}\right), \qquad j \in \{1, \ldots, m\}$$

Иерархический max-pooling, добавляющий учёт порядка слов:



[Shen et al (ACL 2018); Baseline Needs More Love: On Simple Word-Embedding-Based Models]

Альтернативные представления

Можно заменить вектор слова v_w на вектор близостей слова w со всеми словами из W (или только с самыми частотными словами):

$$v_w^{new} = v_w V^T, \qquad V = \left[v_1^T, \dots, v_{|W|}^T\right] \in \mathbb{R}^{|W| \times m}$$

Можно провести кластеризацию матрицы эмбеддингов и использовать вектор близостей слова с центрами кластеров.

Можно в качестве v_w использовать случайный вектор $\in \mathbb{R}^m$

Почему это может работать?

Альтернативные представления

Можно заменить вектор слова v_w на вектор близостей слова w со всеми словами из W (или только с самыми частотными словами):

$$v_w^{new} = v_w V^T, \qquad V = \left[v_1^T, \dots, v_{|W|}^T\right] \in \mathbb{R}^{|W| \times m}$$

Можно провести кластеризацию матрицы эмбеддингов и использовать вектор близостей слова с центрами кластеров.

Можно в качестве v_w использовать случайный вектор $\in \mathbb{R}^m$

Почему это может работать?

Такое представление будет схоже с использованием one-hot векторов с дополнительным шумом.

Задача классификации документов

Дана коллекция документов D, для каждого документа $d \in D$ известна метка класса $y_d \in C$ – множеству классов

Найти для любого документа d его метку класса y_d

Метрики качества:

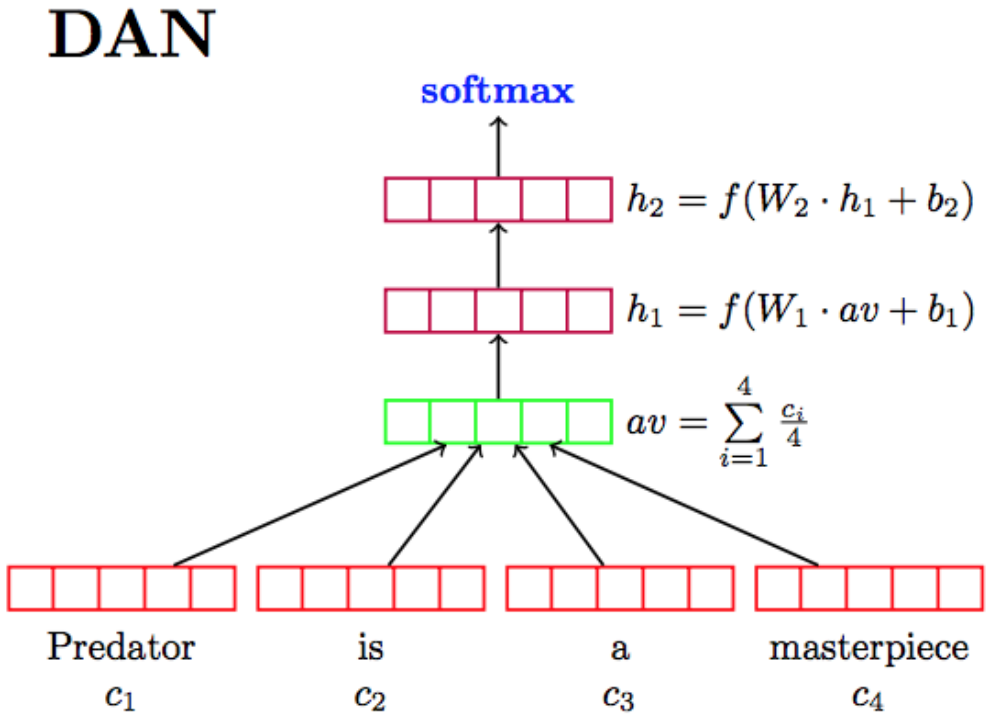
- accuracy (точность) классификации
- **бинарная:** precision (точность), recall (полнота), f-score (f-мера)
- **многоклассовые:** микро/макро-усреднения
- **бинарная (скоринг):** AUC ROC, logloss

Модель Deep Averaging Network

1. Усредняем эмбеддинги слов документа
2. Применяем последовательно несколько feed-forward (линейный + активация) слоёв
3. На выходе применяем softmax

Можно обучать эмбеддинги вместе с моделью или использовать предобученные.

На обучении можно использовать Word Dropout: случайно удаляем некоторые слова при усреднении.



Iyyer et al (ACL 2015); Deep Unordered Composition Rivals Syntactic Methods

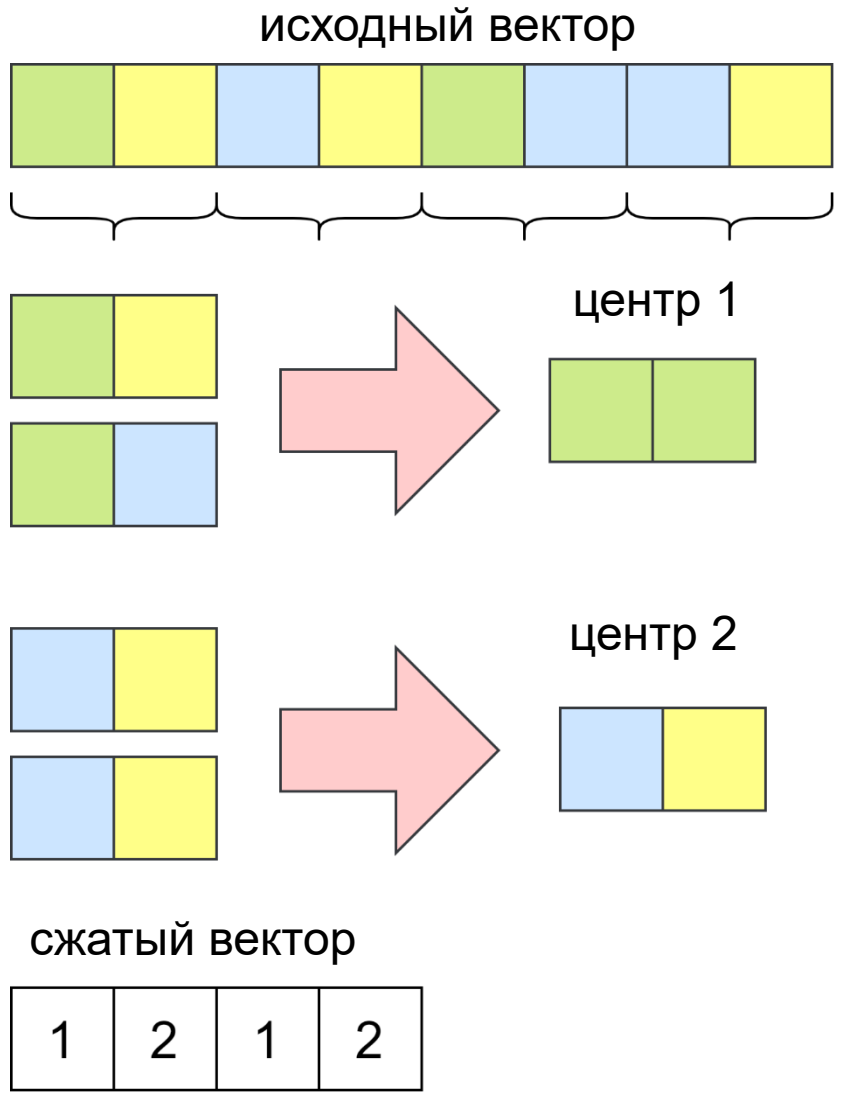
Модель FastText Classifier

- По структуре идентична DAN с одним слоем
- На входе модели слова, n-граммы слов и символов
- Для уменьшения размерности применяется hashing trick
- При большом количестве классов вместо feed-forward слоя с софтмаксом используется иерархический софтмакс на множестве классов
- Есть встроенная процедура подбора гиперпараметров (не всегда работает лучше чем ручной подбор)
- Есть встроенная процедура сжатия эмбеддингов

Joulin et al (2016); Bag of Tricks for Efficient Text Classification

Сжатие FastText: product quantization

1. Каждый вектор делится на части из двухмерных векторов.
2. Двухмерные вектора кластеризуются при помощи K-means
3. Каждый двухмерный вектор заменяется на номер центра его кластера
4. Дополнительно удаляем все представления с малой нормой



Joulin et al (2016); FastText.zip: Compressing text classification models

Задача поиска близких документов (без учителя)

Дано: коллекция документов $D = \{d_1, \ldots, d_N\}$

Найти: близкие (релевантные) документы $d \in D$ для пришедшего нового документа q

Метрика качества: любые метрики из задачи ранжирования (reciprocal rank, precision@k, average-precision@k)

- Выход модели – упорядоченное множество $D' \subset D$ (выдача), чем выше элемент, тем он релевантнее для q
- Если на обучении доступно множество близких пар документов (d, d'), то задача решается в формате обучения с учителем

Поиск на основе эмбеддингов

1. Строим представления всех документов из D
2. Строим представление для документа-запроса q
3. Ищем в D ближайшие документы к q

В качестве меры близости можно использовать:

- скалярное произведение – $\langle v_d, v_q \rangle$
- косинусную близость – $\frac{\langle v_d, v_q \rangle}{\|v_d\|^2 \, \|v_q\|^2}$

Поиск на основе эмбеддингов: анализ

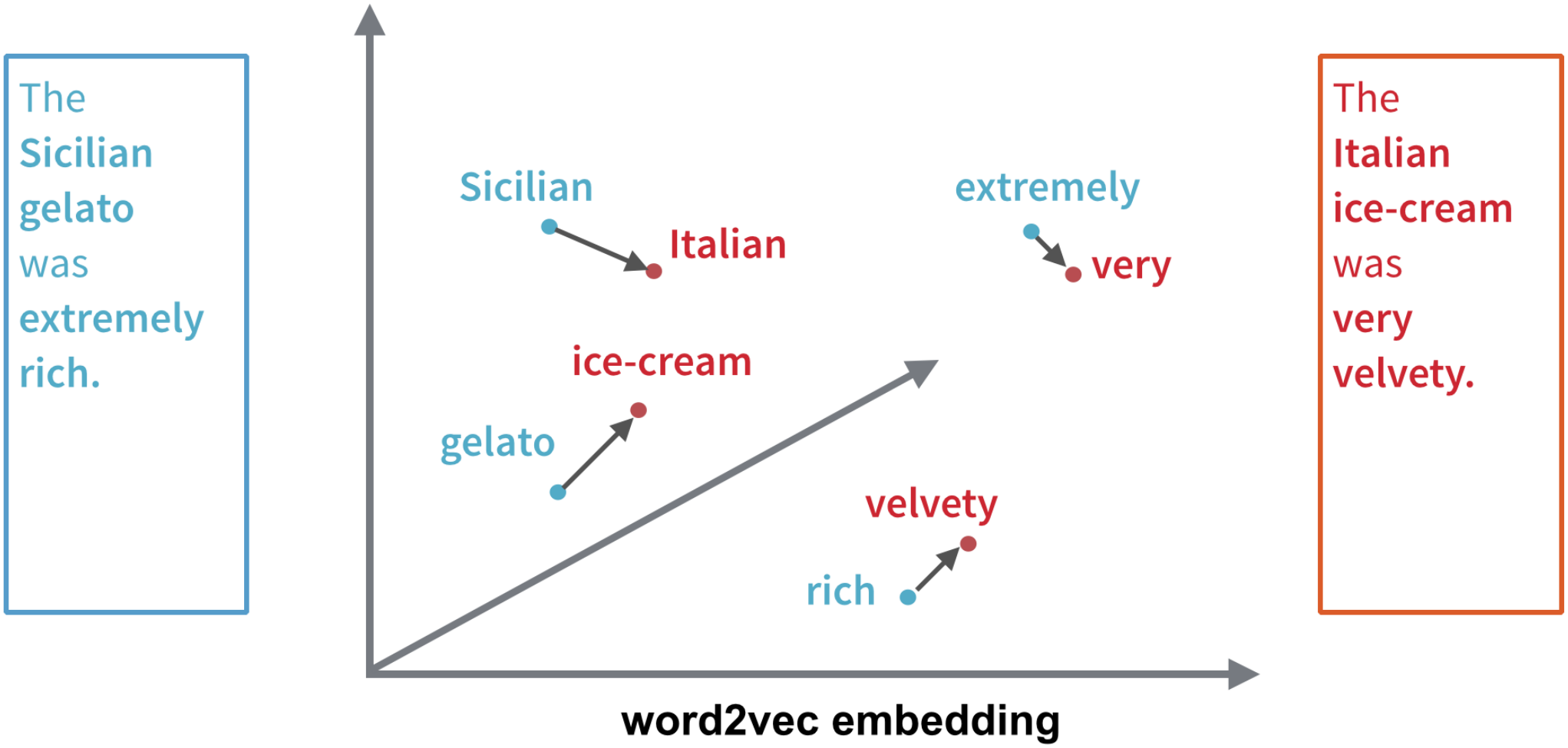
Пусть v_d задаётся средним эмбеддингов, а близость скалярным произведением:

$$\langle v_d, v_q \rangle = \left\langle \frac{1}{|d|} \sum_{w \in d} v_w , \frac{1}{|q|} \sum_{u \in q} v_u \right\rangle = \frac{1}{|d||q|} \sum_{w \in d} \sum_{u \in q} \langle v_w, v_u \rangle$$

Если мы сравниваем два длинных документа, нужно ли учитывать всевозможные попарные близости слов?

Идея. Каждому слову в одном документе сопоставить слово в другом документе и суммировать близости только между соответствующими словами.

Word Mover's Distance (WMD): идея



Kusner et al (JMLR 2015); From Word Embeddings To Document Distances

Word Mover's Distance: определение

Расстояние WMD задаётся как решение оптимизационной задачи:

$$\begin{cases} WMD(d,q) = \min\limits_{T_{wu} \geq 0} \sum\limits_{w \in W'} \sum\limits_{u \in W'} T_{wu} \rho(w,u) \\ \sum\limits_{w \in W'} T_{wu} = n_{uq}\,, \qquad \forall u \in W' \\ \sum\limits_{u \in W'} T_{wu} = n_{wd}\,, \qquad \forall u \in W' \end{cases}$$

n_{wd} – количество появлений слова w в документе d

$T \in \mathbb{R}^{|W'| \times |W'|},\ W' = set(d) \cup set(q) \ \subset W$

Word Mover's Distance на практике

- Перед подсчётом следует исключить из предложений стоп-слова
- Можно заранее предпосчитать $\rho(u, w)$, чтобы уменьшить количество вычислений
- Сложность вычисления WMD: $O(WMD) = O(|W'|^3 \log |W'|)$
- Для понижения сложности можно использовать Relaxed WMD, имеющей сложность $O\left(|W'|^2\right)$

  $WMD_{less}(d, q)$ – решение задачи без второго ограничения

  $RelaxedWMD(d, q) = \max(WMD_{less}(d, q) + WMD_{less}(q, d))$

Полезные ссылки

- Gensim — пакет, позволяющий легко работать с различными моделями эмбеддингов (в том числе учить с нуля)
- fasttext — библиотека для обучения fasttext эмбеддингов с нуля
- Wikipedia2Vec — эмбеддинги для разных языков
- RusVectores — сайт с эмебеддингами для русского языка
- StarSpace — любопытная библиотека/модель, позволяющая учить эмбеддинги под конечную задачу

Итоги занятия

- Существует два подхода к оцениванию представлений: intrinsic и extrinsic.
- Extrinsic подход лучше работает на практике
- Представление документа можно задавать как агрегацию эмбеддингов входящих в него слов
- При классификации документов можно использовать полносвязные сети
- При поиске близких документов можно использовать метод WMD

Бонус!

Если успели до конца лекции

Модель Distributed Memory (paragraph2vec)

Distributed Memory – обобщение модели CBOW для построения представления документа.

$$\mathcal{L} = \sum_{d \in D} \sum_{i=1}^{n_d} \log p(w_i | C(i), d) \to \max_{U,V,\Theta}, \qquad U, V \in \mathbb{R}^{|W| \times m}, \Theta \in \mathbb{R}^{|D| \times m}$$

$C(i) = \{w_{i-k}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+k}\}$ – локальный контекст w_i

Чтобы оценить вероятность, вычисляем вектора контекста и применяем к нему линейный слой с softmax активацией:

$$v_d^{-i} = \frac{1}{2k+1}\left(\sum_{w \in C(i)} v_w + \theta_d\right)$$

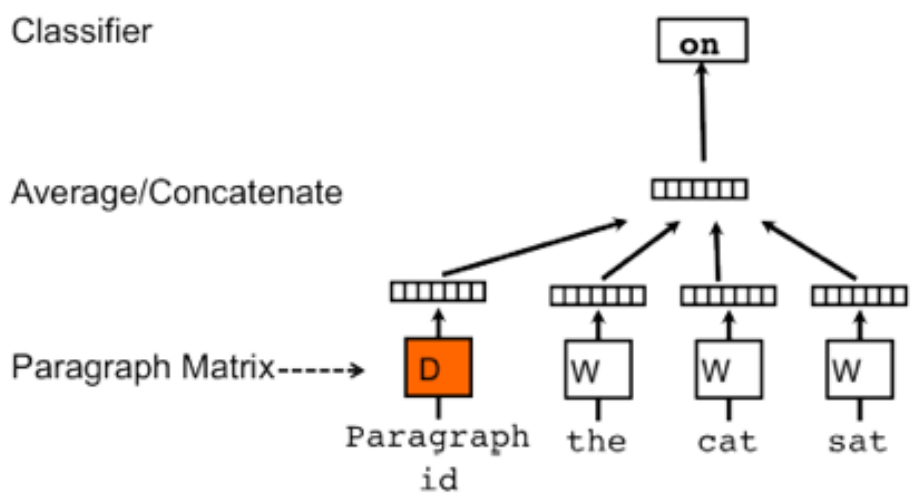
$$p(w | C(w_i)) = softmax_w\left(U v_d^{-i}\right)$$

Модель Distributed Bag of Words (paragraph2vec)

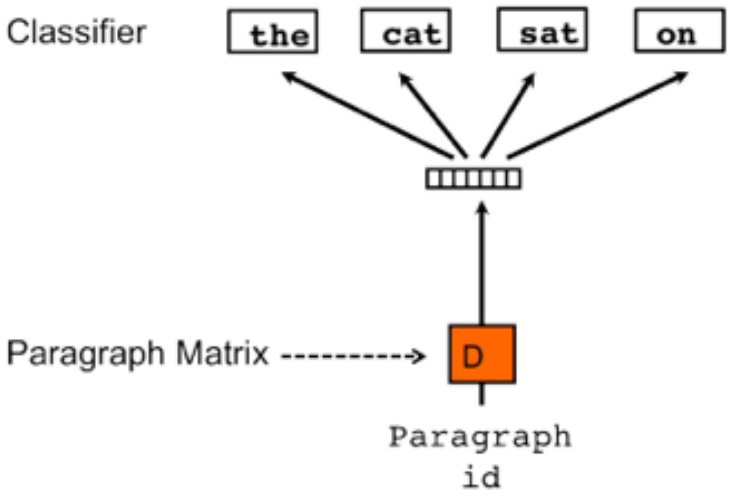
Distributed Bag of Words– обобщение модели Skip-gram для построения представления документа.

$$\mathcal{L} = \sum_{d \in D} \sum_{c \in d} \log p(c|d) \to \max_{U,\Theta}, \qquad U \in \mathbb{R}^{|W| \times m}, \qquad \Theta \in \mathbb{R}^{|D| \times m}$$

$$p(c|d) = softmax_c(U\theta_d)$$



DM (Distributed Memory)



DBOW (Distributed Bag of Words)

Стоит ли использовать paragraph2vec?

Скорее нет, чем да.

1. Результат часто оказывается хуже чем усреднение эмбеддингов
2. Нет нормального способа получить представление для документа не из коллекции