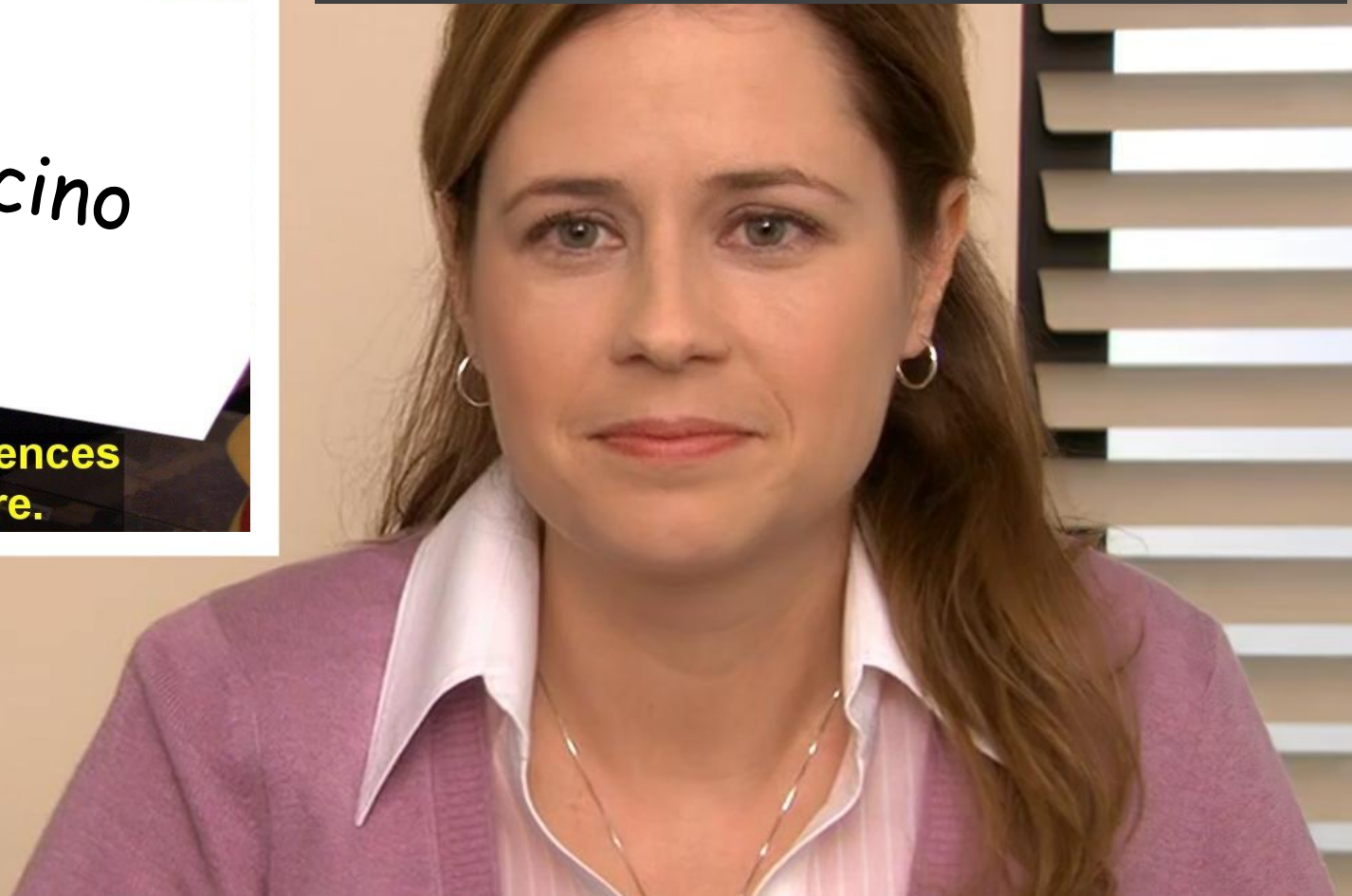


Word2vec + cosine similarity

Espresso

Cappuccino

Corporate needs you to find the differences
between this picture and this picture.



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

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 эмбедингов и **сравниваем** качество.

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

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 ассессорам.

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

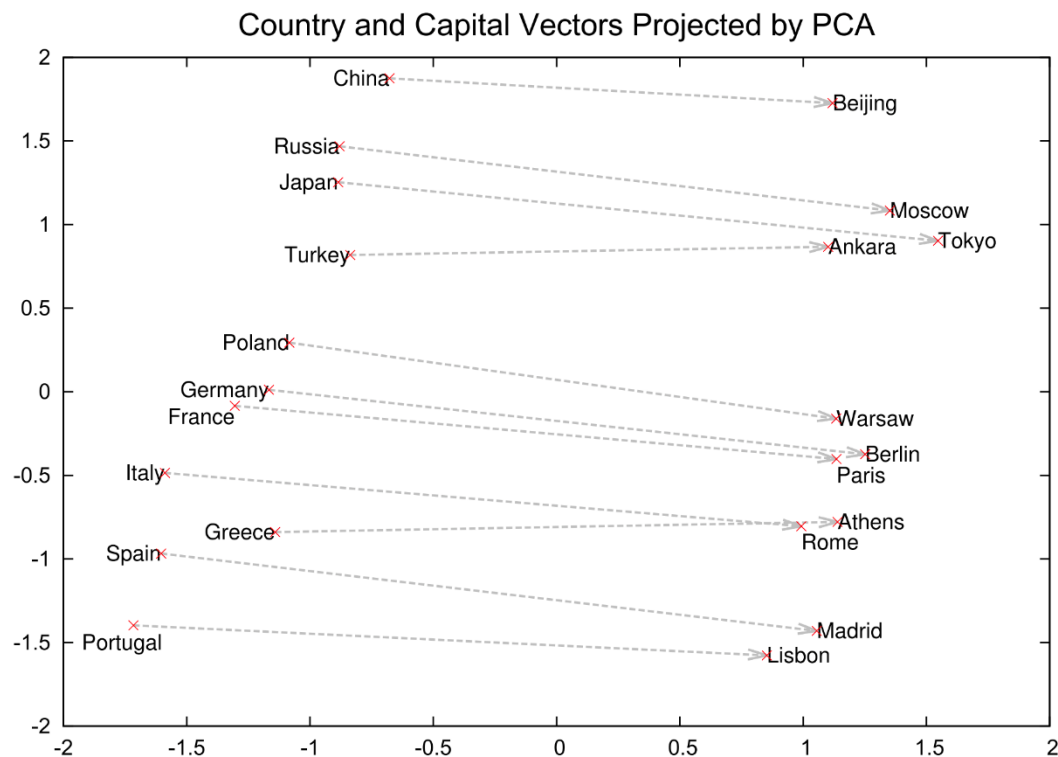
Модель. Находим самое близкое слово к $v(w_3) - v(w_1) + v(w_2)$ кроме самих слов w_1, w_2, w_3

Метрика. Доля правильно найденных слов

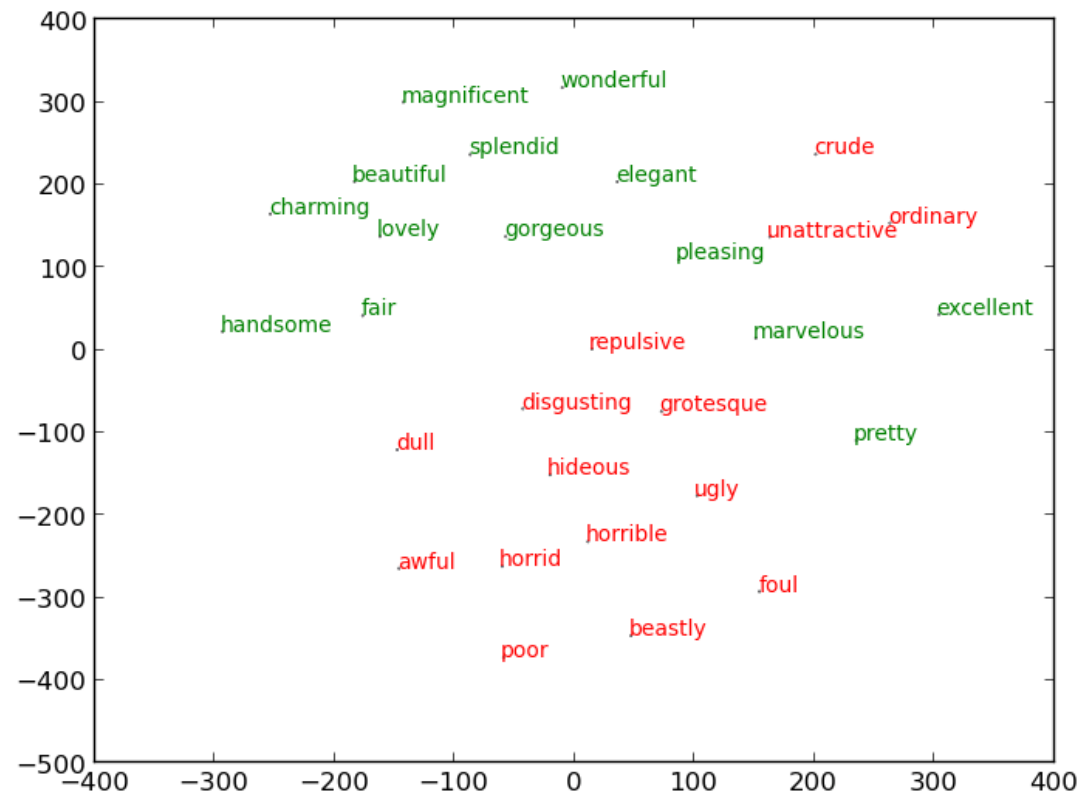
Семантика: $v(king) - v(boy) + v(girl) \approx v(queen)$

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

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

Приложения

Поиск близких документов и
классификация

Как можно использовать эмбединги?

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 частых слов, можно попробовать взвешенное усреднение (сумму):

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

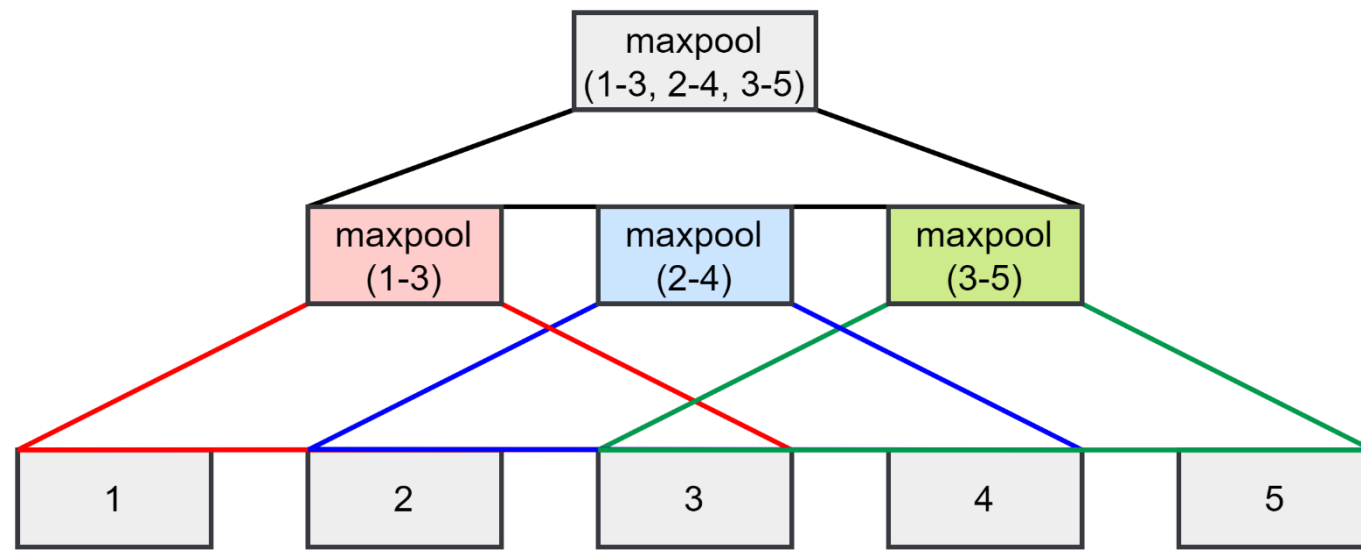
В качестве α_{wd} часто используют iDF значения.

Векторное представление документа

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

$$v_{d,j} = \max(\{v_{w,j} \mid w \in d\}), \quad j \in \{1, \dots, m\}$$

Иерархический max-pooling, добавляющий учёт порядка слов:



Альтернативные представления

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

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

Можно провести кластеризацию матрицы эмбедингов 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):

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

Можно провести кластеризацию матрицы эмбедингов 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:

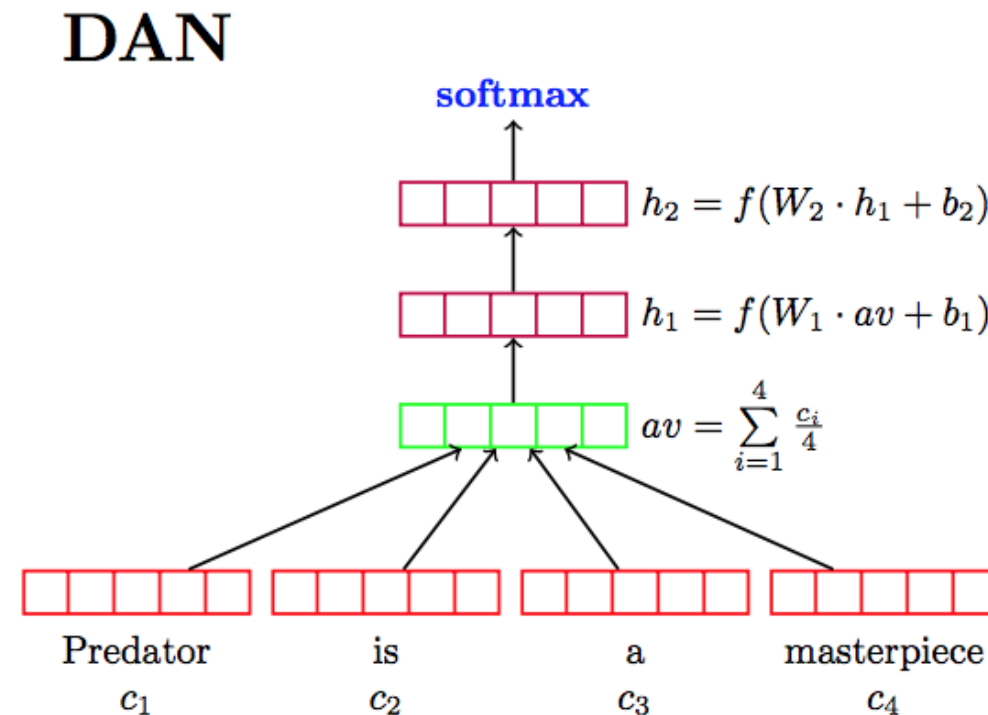
- accuracy (точность) классификации
- **бинарная**: precision (точность), recall (полнота), f-score (f-мера)
- **многоклассовые**: микро/макро-усреднения
- **бинарная (скоринг)**: AUC ROC, logloss

Модель Deep Averaging Network

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

Можно обучать эмбединги вместе с моделью или использовать предобученные.

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

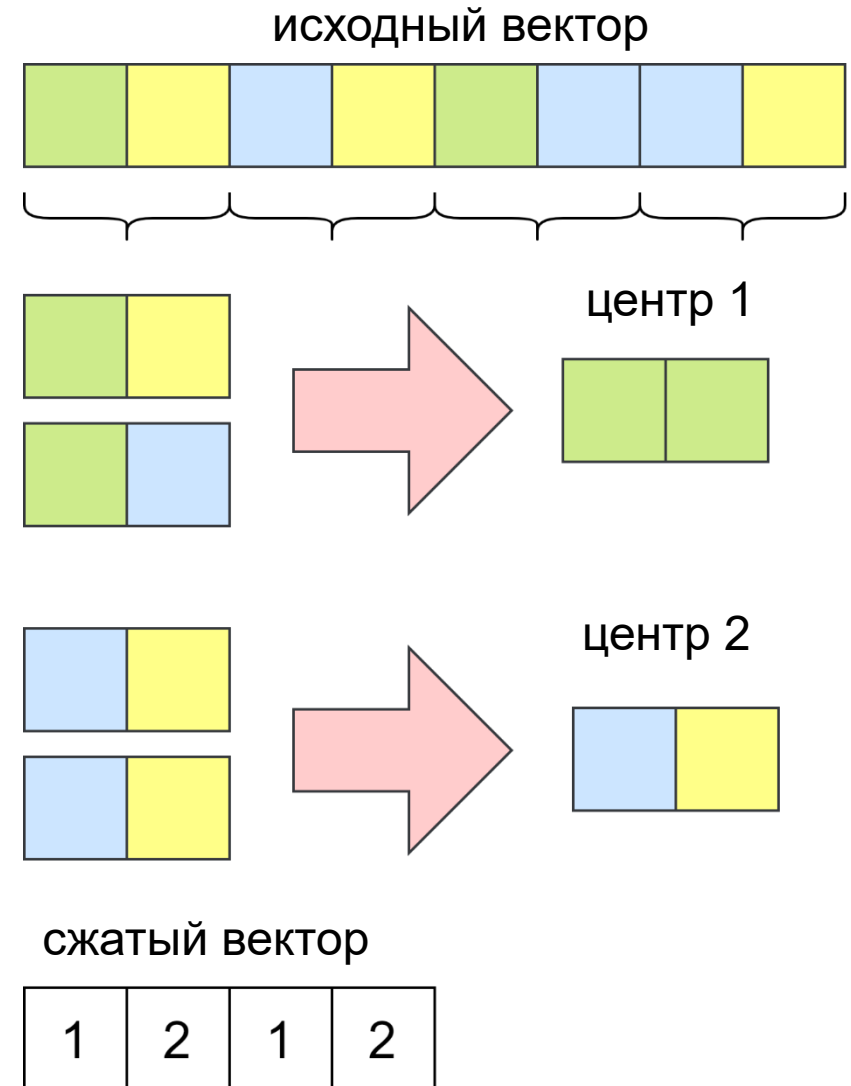


Модель FastText Classifier

- 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

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



Задача поиска близких документов (без учителя)

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

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 эмбедингов

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 эмбедингов: анализ

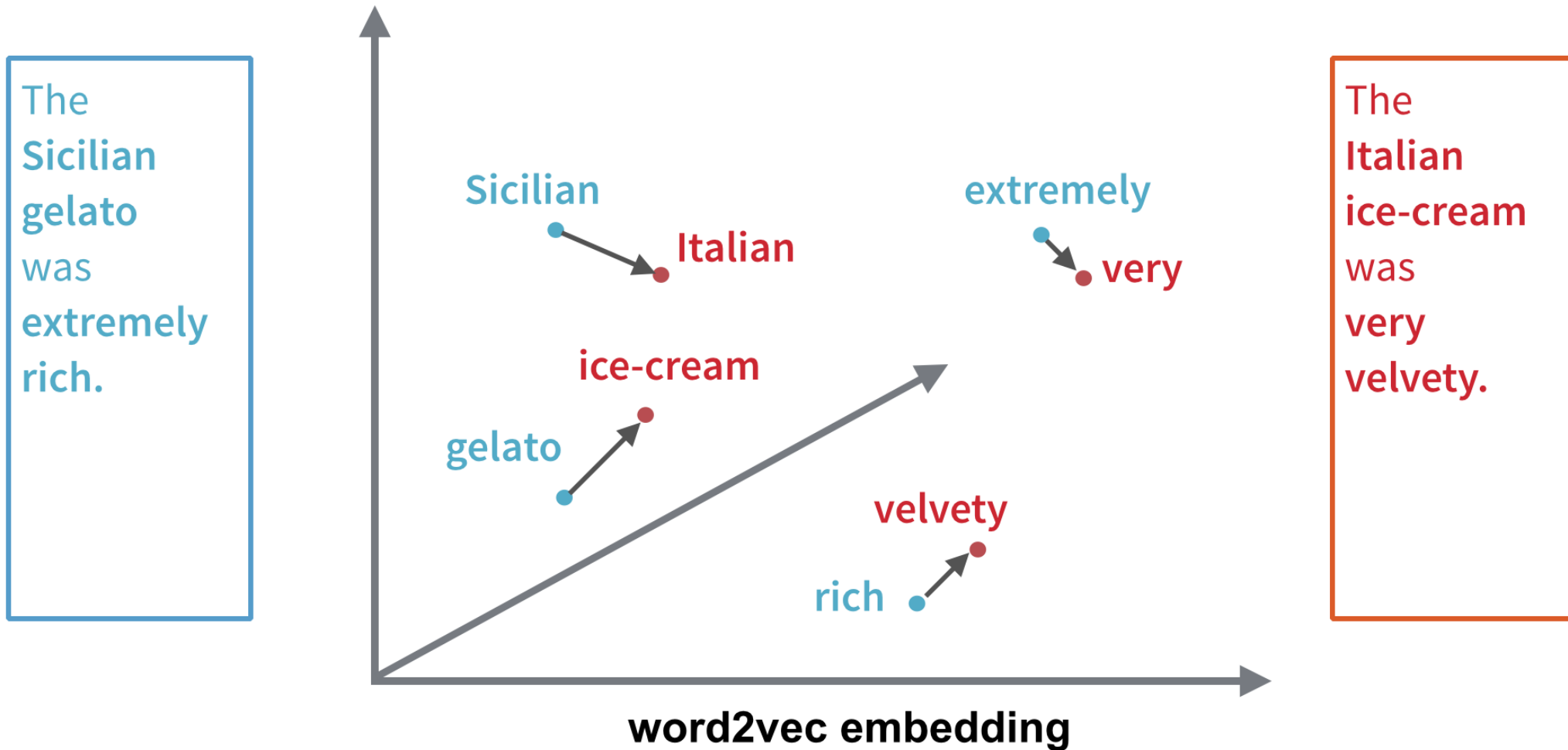
Пусть v_d задаётся средним эмбедингов, а близость скалярным произведением:

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

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



Word Mover's Distance: определение

Расстояние WMD задаётся как решение оптимизационной задачи:

$$\left\{ \begin{array}{l} WMD(d, q) = \min_{T_{wu} \geq 0} \sum_{w \in W'} \sum_{u \in W'} T_{wu} \rho(w, u) \\ \sum_{w \in W'} T_{wu} = n_{uq}, \quad \forall u \in W' \\ \sum_{u \in W'} T_{wu} = n_{wd}, \quad \forall w \in W' \end{array} \right.$$

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

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

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 $O(|W'|^2)$

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

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

Итоги занятия

- Существует два подхода к оцениванию представлений: intrinsic и extrinsic.
- Extrinsic подход лучше работает на практике
- Представление документа можно задавать как агрегацию эмбедингов 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.

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

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

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

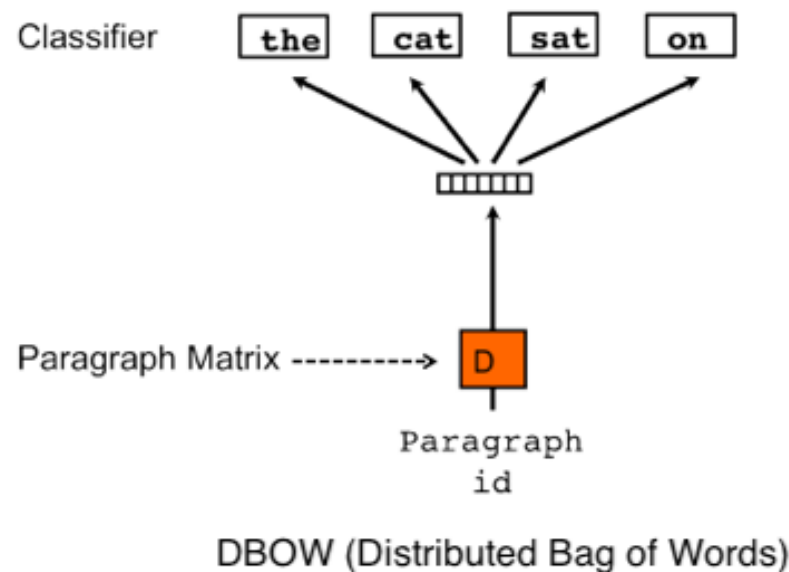
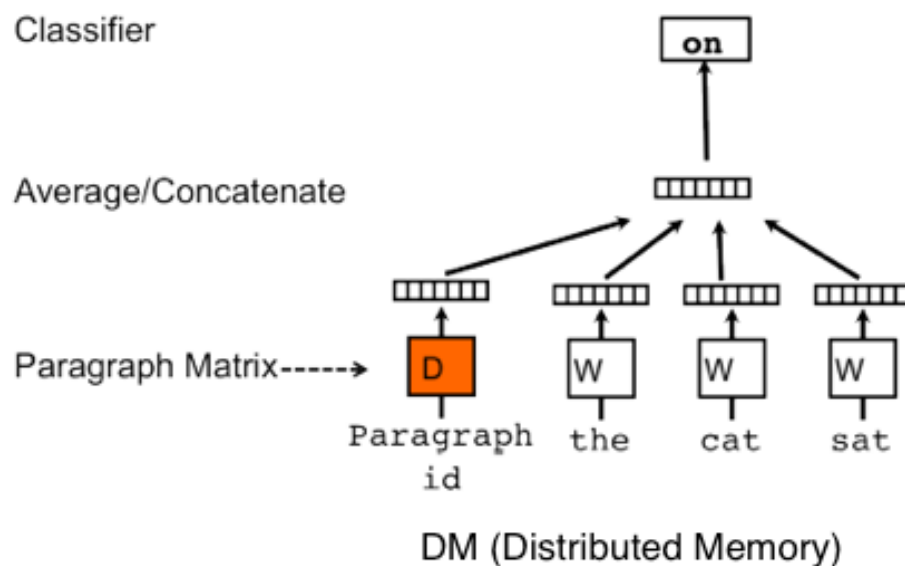
$$v_d^{-i} = \frac{1}{2k+1} \left(\sum_{w \in C(i)} v_w + \theta_d \right)$$
$$p(w | C(w_i)) = \text{softmax}_w (U v_d^{-i})$$

Модель Distributed Bag of Words (paragraph2vec)

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

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

$$p(c|d) = \text{softmax}_c(U\theta_d)$$



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