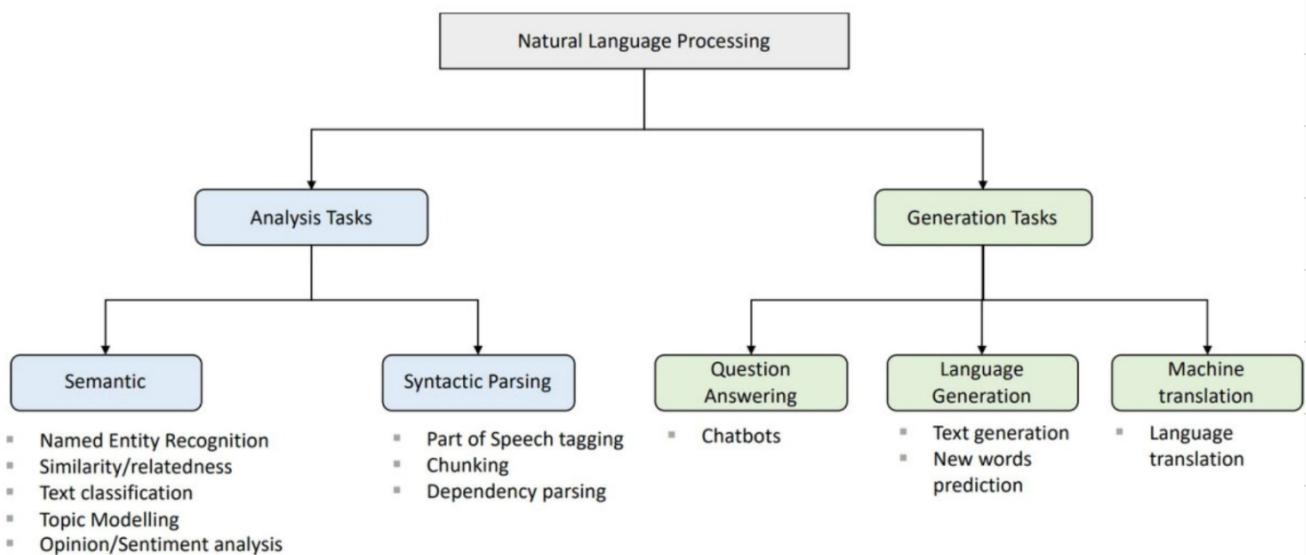


Лекция 8. Индексации слов, генеративная шпонометра, word2vec

Примеры основных задач:

- Дискриминативное (классификация, распознавание сущностей, похожесть текстов...)
- Генеративное (генерация текстов, условная генерация: перевод, чатботы)



Различные модели

Дана выборка документов $\{d_1, \dots, d_n\}$. Каждый документ –
множество содержимого слова w_1, \dots, w_L , где L_d – число слов в документе d .
Важно заметить – слова дискретное и задан словарем W все возможных слов.

Задача: Оценить вероятность след. слова в предложении.

$$P(w|w_1, \dots, w_k)$$

Меморизаци

$$\mathcal{L}(D_{\text{test}}) = \prod_{d \in D_{\text{test}}} \prod_{n=1}^{|L_d|} p(w_n | w_1, \dots, w_{n-1}) - \text{правдоподобие}$$

$$\mathcal{L}(D_{\text{test}}) = \prod_{d \in D_{\text{test}}} \left\{ \prod_{n=1}^{|L_d|} p(w_n | w_1, \dots, w_{n-1}) \right\}^{-\frac{1}{|L_d|}} - \text{нормализация}$$

Приложения

- Исправление опечаток и грамматических ошибок
- машинный перевод: генерировать наиболее вероятную последовательность слов на языке, на котором нужно перевести, при условии, что мы исходим из исходного языка

Классические подходы в языковых моделях

Не обект. смотреть на "слова" как на единичные конструкции "предложений"

N-грамматическое модели

Предположение — слова в начале текста не особо влияют на слова в конце. Есть какой-то контекст, в рамках которого слова влияют друг на друга

$$p(d) = p(w_1, \dots, w_N) \prod_{n=N+1}^L p(w_n | w_{n-1}, \dots, w_1) \approx$$

$$\approx p(d) = p(w_1, \dots, w_N) \prod_{n=N+1}^L p(w_n | w_{n-1}, \dots, w_N) \sim$$

Распределения можно оценивать частотно по всем *N*-граммам в корпусе документов:

$$p(w | w_1, \dots, w_k) = \frac{C(w_1, \dots, w_k, w)}{C(w_1, \dots, w_k)}$$

Какие проблемы?

- всё такие большие размерности
- неизвестные слова никак о вер-ть
- Чем больше *N*, тем лучше учитываются контекст
- Предложения могут быть средой, но отдельное *n*-грамма — осмыслено
- оценка первого слова

Помощные идеи решения

- (1) Добавлять в начало предложений $\langle \text{start} \rangle$, чтобы лучше моделировать первые первых слов.
- (2) Направление нормировкой

$$p(w | w_1, \dots, w_n) = \frac{C(w_1, \dots, w_n, w) + d}{C(w_1, \dots, w_n) + d | w|}, \text{ где } d\text{- мало}$$

Katz backoff. Основная идея: если не встречали $(k+1)$ -грамму, то шанс произвести 'окном'

$$C(w_{n-k}, \dots, w_n, w) = \begin{cases} C(w_{n-k}, \dots, w_n, w), \beta(w_{n-k}, \dots, w_n) & \text{если } C(w_{n-k}, \dots, w_n) > 0 \\ C(w_{n-k+1}, \dots, w_n, w), d(w_{n-k}, \dots, w_n) & \text{иначе} \end{cases}$$

Общая задача построения индексиков

Дано: $D = \{w_1, \dots, w_{N-d}\}_{d=1}^{d=N}$ — корпус текста, $w_i \in W$ — слова в невозможных слов (единичную конструкции предложений)

Найти векторное представление $v_w \in \mathbb{R}^m$ для w , где $m \ll |W|$

Хотим:

1. Соответствие близости по смыслу слов к близости по расстоянию между векторами

2. Интерпретируются арифм. операции над словами в пр-ве энSEMBЛИгов. То есть операции, сохраняющие "смысл" слов
Если получится, то решим задачи:

1. Поиск синонимов слов, синонимов и т.п.
2. Получение представления докуентиста, которое будет использоваться в других задачах машинного обучения
3. Использование в качестве фиксированного предмета венчия в сложной архитектуре
4. Использование для инициализации представлений в сложной архитектуре

Гипотеза дескрибутивности

1. Основная гипотеза:

Слова, совместно употребляемые с одними и теми же словами, имеют сходное значение.

2. Основная гипотеза:

Слова характеризуются словами, с которыми они сопротивляются

* В русской языке это слабое выражение, т.к. структура предложений часть не фиксирована

Histogram 1: Count-based

Цель: сформировать в векторах смысл слов и понятий, кроме того мы хотим уменьшить разнородность

$$p(w) \approx \frac{n_w}{N} - \text{частота слова } w$$

$$p(w, v) \approx \frac{n_{w,v}}{N^2} - \text{частота пары } w \text{ и } v \text{ вместе}$$

один-один слово в тексте

Взаимная информация двух слов (pointwise mutual information / PMI), что по смыслу отвечает на вопрос: "На сколько одно слово влияет на другое?"

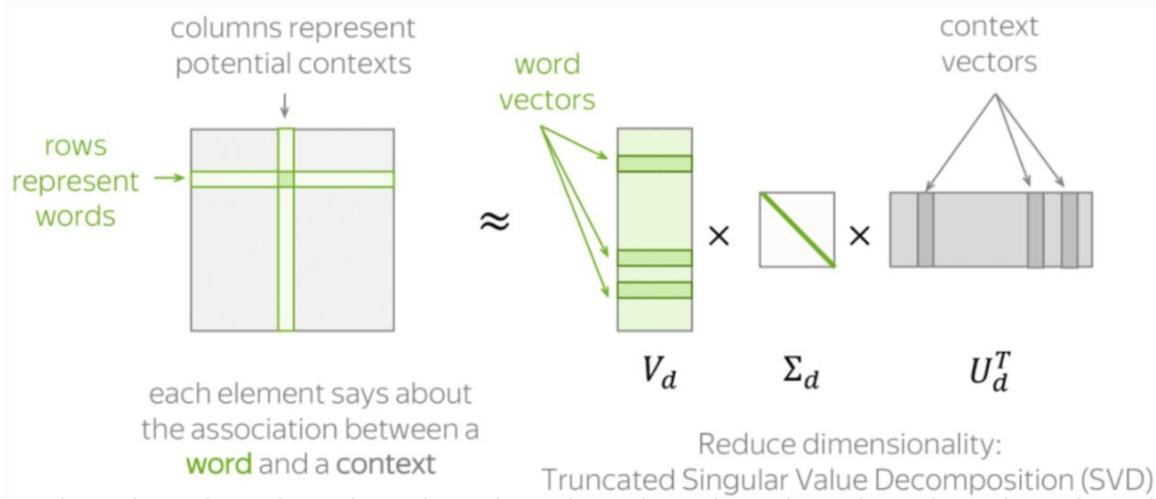
$$\text{PMI} = \log \frac{p(w, v)}{p(w)p(v)} = \log \frac{n_{w,v}}{p(w)p(v)} = \log \frac{n_{w,v}}{n_w \cdot n_v}$$

Позитивная взаимная информация:

$$\text{PPMI}(w, v) = \max(0, \text{PMI}(w, v))$$

Составим матрицу $X \in \mathbb{R}^{|W| \times |W|}$ встречаний, где $X_{w,v} = \text{PPMI}(w, v)$

Мы получили очень большую разреженную матрицу, в которой содержится полезная информация, теперь давайте уменьшим ее размер. Для этого воспользуемся методами понижения размерности, например, SVD разложением.



Glove: Global Vectors for Word Representation

Зададим матрицы U, V , которые будут содержать вектора слов и контекста, тогда мы можем решить

след. оптимизационную задачу:

$$\mathcal{L} = \sum_{w \in W} \sum_{c \in W} F(n_{w,c}) \left(\langle U_w^T V_c + b_w + b_c \rangle - \log n_{w,c} \right)^2$$

$F(n) = \begin{cases} \frac{(n_{w,c})^\alpha}{n_{\max}} & \text{если } n_{w,c} < n_{\max} \\ 1 & \text{иначе} \end{cases}$

дано α для обучения векторов
связь контекстом
сигн. произв.
разширяет вектор пары
от невидимых

\min_{U, V, b_w, b_c} $\text{хотя бы близки с тем же значением}$

$$\mathcal{L} = \frac{3}{4}, \quad n_{\max} = 100, \quad \text{ипер-параметр статьи.}$$

* в статье более подробно, советуют прочитать для получения интуиции

Слово функционирует, чем чаще встречаются слова и с в одном контексте, тем выше $F(n_{w,c})$, а значит тем выше достоверность $v_w^T v_c + b_w + b_c \leq \log n_{w,c}$.

На практике используется, но работает хуже более простых методов.

Преимущества:

- Неплохое качество в некоторых задачах (но предустановлено)
- Маленькая размерность

Недостатки:

- Нет хорошего механизма обработки новых слов
- Необходимо собирать огромную матрицу соответствий для обучения

Наш недостаток: масштабирование.

Ноги 2: Prediction-based (word2vec)

Идея: будем обучать модель "выводить" имена между якоря. Тогда есть два основных подхода:

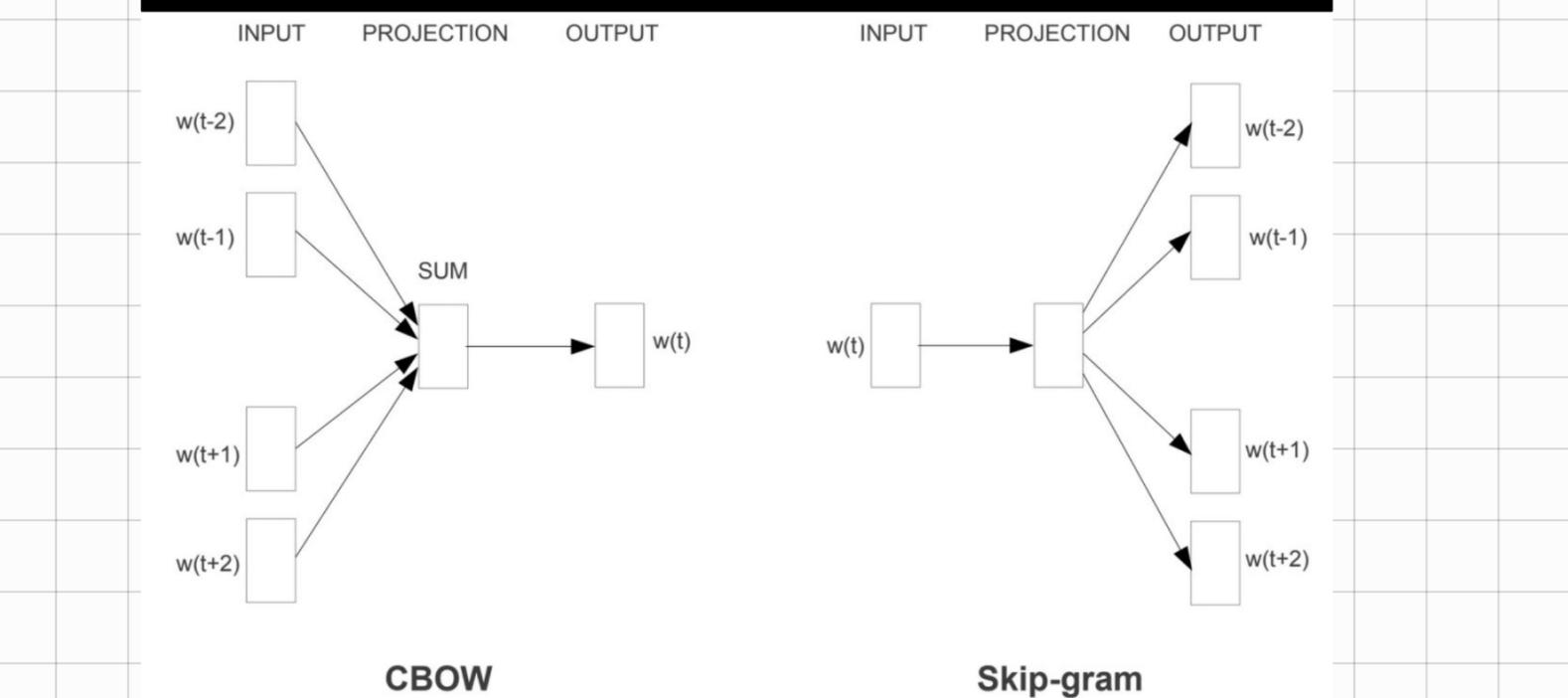
- Модель CBOW - по словам контекста необходимо предсказать центральное слово
- Модель Skip-gram - по центральному слову предсказать остальные слова контекста

Модель CBOW (continuous bag of words)

Если слово характер. соседними словами, то можем обучить предсказывать слово по контексту

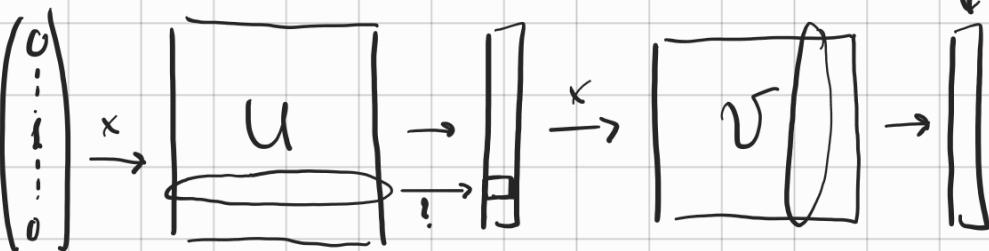
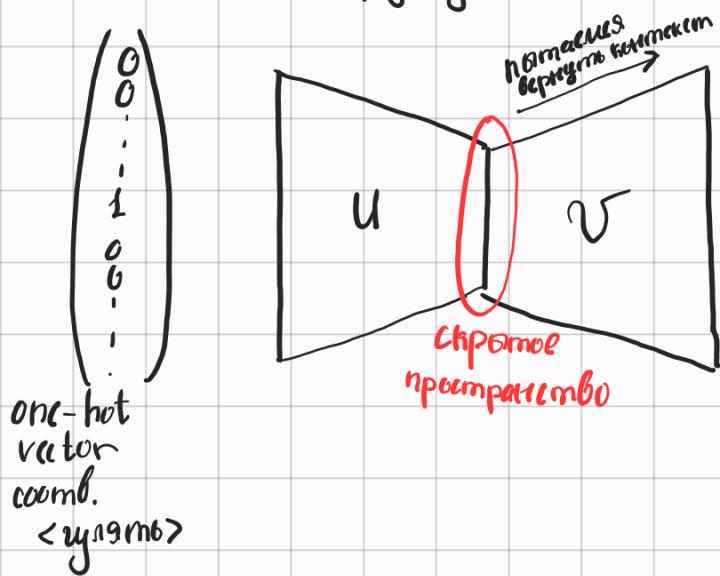
Модель Skip-gram

— // —, можем предсказывать контекст
по слову.



Я пошел гулять на улицу

предсказ. по нему



Изображенные слова будем называть либо строкой u либо столбцом v

Формулировка:

$$\text{CBOW}: \sum \log p(w_i | C(w_i)) \rightarrow \max_{U, V}$$

$$p(w_i | C(w_i)) = \text{softmax}_{w \in C(w_i)} (U_w v^T)$$

$$v = \frac{1}{|C(w_i)|} \sum_{w \in C(w_i)} V_w$$

сумма строк

Инициализацию, ищем слово, которое близко к контексту ($U_w v^T$)

V_w - признаки о слове, как о контексте

U_w - признаки о слове, как о матрите. Даём векторный склер продукт с близкими контекстами

Skip-Gram:

$$\sum_i^N \sum_{w \in C(w_i)} \log(p(w|w_i)) \rightarrow \max_{U, V}$$

$$p(w|w_i) = \text{softmax}_{w \in W} (U_w V_{w_i}^T)$$

Замечания:

перевод \downarrow в вер-е
пространство

- Skip-gram - обучение редкие слова
- на практике изглаживай

Основное проблема:

- нет возможности добавлять новые слова
- необходимость хранения матриц.

FastText

FastText

Идея — будем строить векторы для частей слов, а не для целых слов.

- Делим слова на n-граммы по буквам:
 $\text{apple} = \langle \text{ap}, \text{ppl}, \text{ple}, \text{le} \rangle$
- Учим векторы для n-грамм;
- Вектор слова получаем как сумму векторов его n-грамм.

Плюсы:

- Можно получить более адекватные эмбеддинги для редких и неизвестных слов;

Недостатки:

- n-грамм может быть очень много. Требуется больше вычислительных ресурсов.

Byte-Pair Encoding (BPE)

Основная идея: будем строить слова иерархически.

1. Изначально слова рассматриваются посимвольно

2. Подсчитываются пары символов: как часто пары символов идут подряд.

3. Находится самая частная пара
 4. Пара символов объединяется в новый символ.
 5. Большой словарь — заканчивается. Ниже — пункт 2
-

AABABCABBAABAC

AA - 2

AB - 4 AB = D

BA - 3

BC - 1

CA - 1

BB - 1

AC - 1

ADDcdbADAC

AD - 2 AD = E

DD - 1

DC - 1

CD - 1

DB - 1

DA - 1

AC - 1

EDCdbeAC

Решает проблему ограниченности словаря и тренир. выборки

Пример: **s-u-b-w-o-r-d**



s-u-b-w-o-r-d



...

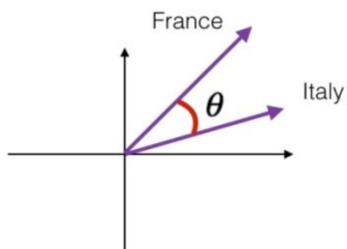


sub-word

Интерпретация эмбеддингов с использованием косинусной меры сходства

$$\text{Cosine Similarity}(u, v) = \frac{u \cdot v}{\|u\|_2 \cdot \|v\|_2} = \cos(\theta)$$

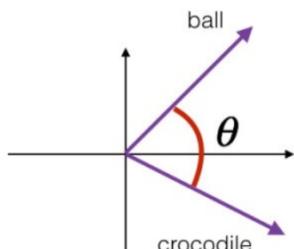
θ -угол между векторами u, v , выраженный как косинус меры
близости



France and Italy are quite similar

θ is close to 0°

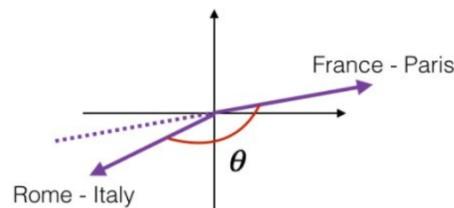
$\cos(\theta) \approx 1$



ball and crocodile are not similar

θ is close to 90°

$\cos(\theta) \approx 0$



the two vectors are similar but opposite
the first one encodes (city - country)
while the second one encodes (country - city)

θ is close to 180°

$\cos(\theta) \approx -1$

Используем нулевое представление для токена

Арифметика в поле эмбеддингов

Хотим интерпретировать смену арифм. операций.

Word2Vec

На векторах word2vec можно проводить векторную арифметику:

$$v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$$

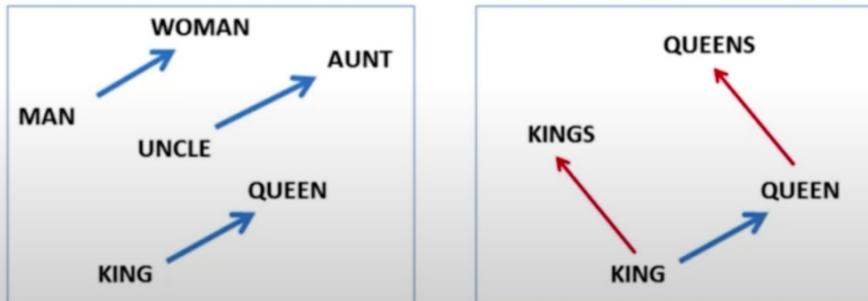
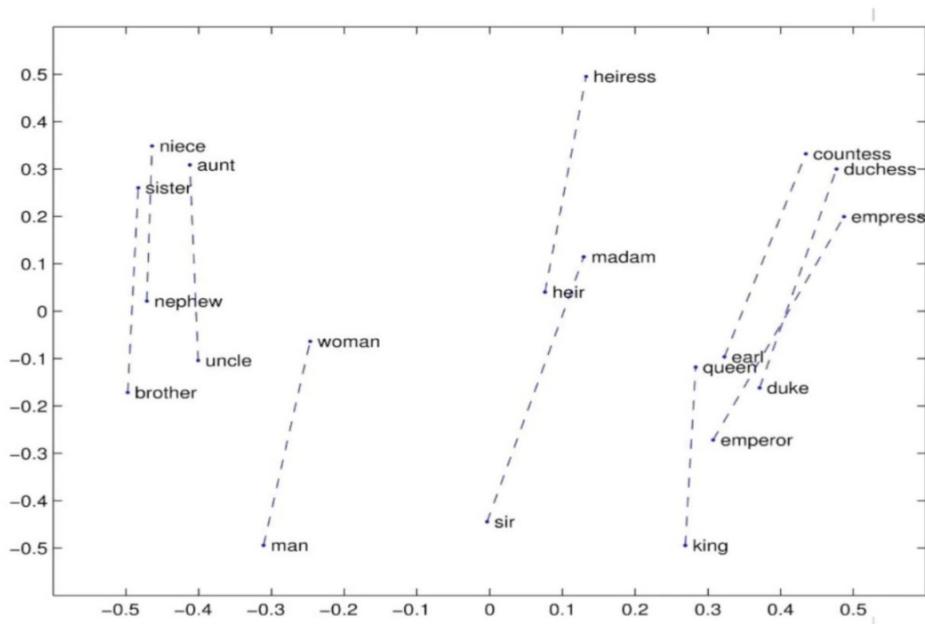


Иллюстрация эмбеддингов после понижения размерности

Word2Vec



вектор разницы мужских и женских слов почти одинаков

Но бенчмарки GloVe не сильно проигрывают даже

экзапоному BERT и ST5-XXL (разница большая но не в 2 раза)

