

Препроцессинг; языковые модели

Екатерина Владимировна Еникеева

13 сентября 2022

Автоматическая обработка естественного языка, лекция 2

План занятия

- Объекты NLP: что обрабатываем
- Токенизация и subword segmentation: как выделить минимальные единицы анализа
- Языковые модели: как оценить вероятность текста

Терминология

- корпус > документ/текст > предложение > токен
- корпус: обучающий / тестовый / валидационный
- словарь – множество всех уникальных токенов корпуса
- **ngram** (n-грамма) — последовательность токенов длиной N:
 - униграмма
 - биграмма
 - триграмма
 - ...
- Language Model (LM)

Предобработка

= *Preprocessing*

- фильтрация текстов (дедупликация и тп)
- удаление нетекстовых элементов, нормализация (кодировка и тд)
- сегментация на предложения
- фильтрация на уровне предложения (например, слишком короткие или длинные)

Токенизация

<https://colab.research.google.com/drive/1ne5HCczQSwCh6m0wj2VJvOA8Rbh7-fxk?usp=sharing>

Sub-word segmentation

Идея: Sehnrich et al. (2016) Neural Machine Translation of Rare Words with Subword Units.

- Проблема размера словаря:
 - именованные сущности, неологизмы, окказионализмы ...
 - компаунды: Abwasser|behandlungs|anlage
'сточные воды | очистка | установка'
- Byte-Pair Encoding: итеративно объединяем самые частотные пары символов

BPE

AABABCABBAABAC

AA - 2

AB - 4 AB = D

BA - 3

BC - 1

CA - 1

BB - 1

AC - 1

ADD^CDBADAC

AD - 2 AD = E

DD - 1

DC - 1

CD - 1

DB - 1

DA - 1

AC - 1

EDCDBEAC

BPE / WordPiece / SentencePiece / ULM
<https://github.com/huggingface/tokenizers>

Пример

(в тетрадке)

Языковая модель

- Language Model: statistical / neural ...
- Модель – упрощенное представление объекта, отражающее определённые его свойства
 - модель синтаксиса
 - модель именных групп noun + adj
 - семантическая модель
 - лексическая семантика
 - ...
- алгоритм + параметры

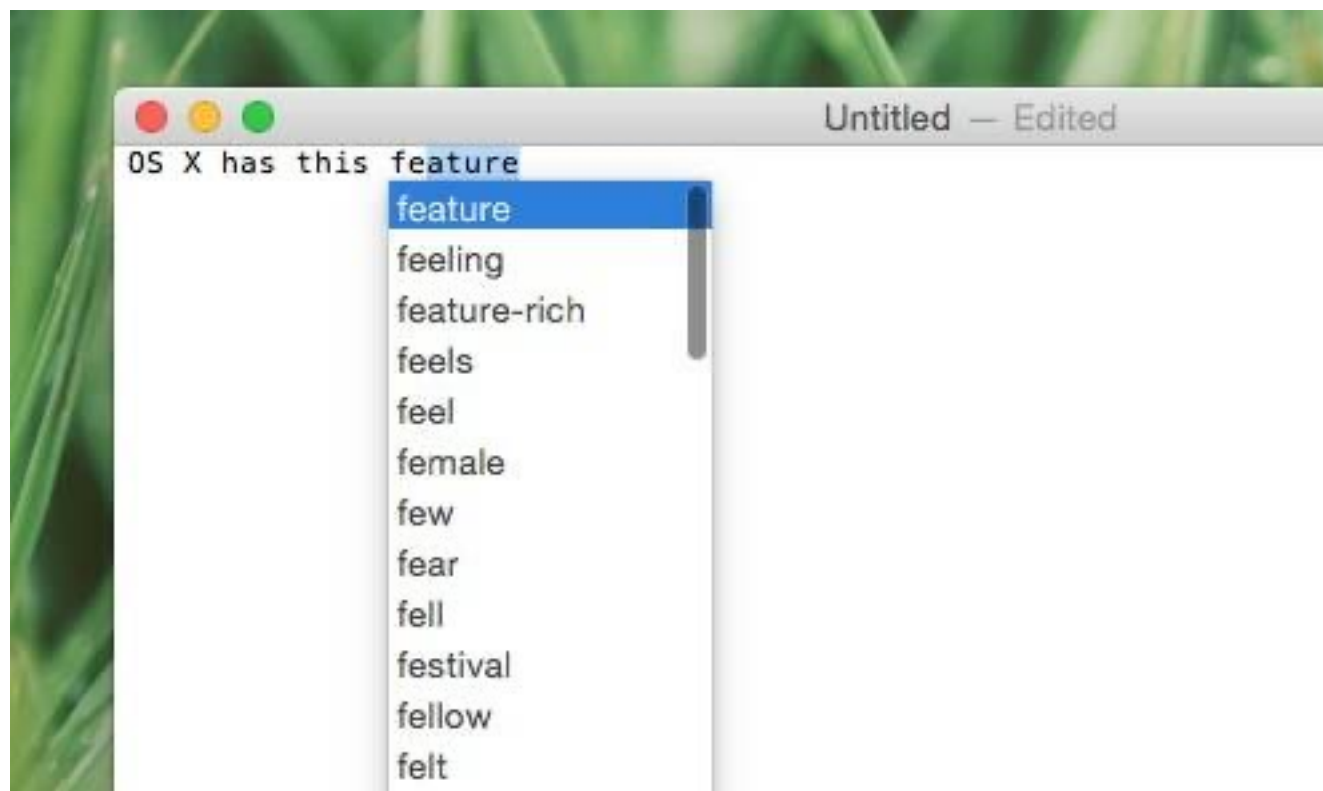
Задача языкового моделирования

- **Цель** – оценить вероятность фразы/предложения/текста
- **Приложения:**
 - машинный перевод
 - распознавание речи
 - исправление опечаток
 - ...

Задача языкового моделирования

- **Цель** – порождение/генерация языковой последовательности
- **Приложения:**
 - саджест (подсказки) в системах ввода
 - машинный перевод
 - диалоговые системы
 - автоматическое реферирование
 - ...

Примеры



Примеры

погода в казани

✕

Поиск

Картинки

Видео

Карты

Маркет

Новости

Переводчик

Кью

Услуги

Исправлена опечатка в слове «казане»


Отменить

Погода в Казани

Яндекс.Погода ...

По часам · На 10 дней · На месяц · Осадки на карте

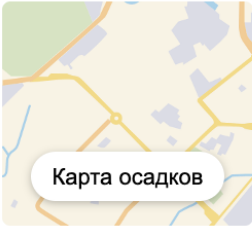
+15°



Пасмурно

4,5 м/с ветер


+15 вечером, +13 ночью



Карта осадков

сегодня


+15



+11

пт 10


+15



+12

сб 11


+15



+7

вс 12

+14



+9

Прогноз
на 10 дней

>

Цель моделирования

- оценка вероятности предложения:

$$P(W) = P(w_1, w_2, w_3, w_4, w_5 \dots w_n)$$

- оценка вероятности следующего слова в предложении:

$$P(w_6 / w_1, w_2, w_3, w_4, w_5)$$

Как оценить вероятность?

- Мы не можем составить корпус, содержащий все возможные предложения языка
- Значит, нужно оценивать небольшие последовательности
- ngrams (n-граммы) – последовательности из n слов

Вероятность слова — ?

Какова вероятность слова *lamb*
в английском языке?

Вероятность слова — ?

Mary has a little lamb .
Mary has a lot of money .

Какова вероятность слова *lamb* в данном корпусе?

Вероятность слова

Оценка вероятности
последовательности w
(слово, n-грамма) в
корпусе C :

$$P(w_i) = \frac{\text{count}(w_i)}{\sum_{w \in C} \text{count}(w)}$$

Mary has a little lamb .
Mary has a lot of money .

$$|C| = 13$$

$$P(\text{*lamb*}) = 1 / 13 \approx 0.077$$

$$P(\text{*Mary*}) = 2 / 13 \approx 0.153$$

NB! Хорошо бы ещё иметь токен «граница предложения» <s>

Как оценить вероятность?

Условная вероятность

$$P(B/A) = P(A,B)/P(A) , \text{ то есть } P(A,B) = P(A)P(B/A)$$

> **chain rule**

В общем случае

$$P(x_1, x_2, x_3, \dots, x_n) = P(x_1) P(x_2/x_1) P(x_3/x_1, x_2) \dots P(x_n/x_1, \dots, x_{n-1})$$

N-gram language model

Предполагаем, что модель обладает Марковским свойством:

вероятность следующего токена зависит только от $k-1$ предыдущих

> Модель порядка k – **k -gram language model**

$$P(w_1 w_2 \dots w_n) \approx \prod_i P(w_i | w_{i-k} \dots w_{i-1})$$

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-k} \dots w_{i-1})$$

N-gram language model

unigram – учитываем только текущее слово

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i)$$

bigram – учитываем предыдущее слово

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-1})$$

trigram = учитываем 2 предыдущих слова

$$P(w_i | w_1 w_2 \dots w_{i-1}) \approx P(w_i | w_{i-1} w_{i-2})$$

Условная вероятность

$$P(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Mary has a little lamb .
Mary has a lot of money .

$$P(\text{*lamb*} \mid \text{*a, little*}) = 1$$

$$P(\text{*little*} \mid \text{*has, a*}) = 1/2$$

$$P(\text{*lot*} \mid \text{*has, a*}) = 1/2$$

Оценка вероятности предложения

Пример 3-граммной модели:

$P(\text{"Mary has a little lamb ."}) =$

$= P(\text{Mary} \mid \langle s \rangle, \langle s \rangle) \times P(\text{had} \mid \langle s \rangle, \text{Mary}) \times$

$\times P(\text{a} \mid \text{Mary}, \text{has}) \times P(\text{little} \mid \text{has}, \text{a}) \times$

$\times P(\text{lamb} \mid \text{a}, \text{little}) \times P(. \mid \text{little}, \text{lamb})$

Сглаживание

smoothing

> Что, если какой-то последовательности не будет в корпусе? Например:

count(a, little, lamb) = N, но count(the, little, lamb) = 0

Попробуем оценивать вероятность меньших частей:

count(a, little)

count(little, lamb)

Немного терминологии

- *Data sparsity*: даже хороший репрезентативный корпус не позволит идеально оценить вероятности
- *Zeros / OOV* – out of vocabulary words могут встретиться в тестовом корпусе

Backoff smoothing

$$P(lamb \mid the, little) = 0$$

> пробуем биграммы

$$P(lamb \mid the)$$

> если снова 0, то пробуем униграммы

$$P(lamb)$$

Какие есть недостатки у такого метода?

Линейная интерполяция (1)

linear interpolation

> Лучше использовать все варианты 1...k-грамм и единообразно для всех подпоследовательностей:

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1 P(w_n|w_{n-2}w_{n-1}) \\ &\quad + \lambda_2 P(w_n|w_{n-1}) \\ &\quad + \lambda_3 P(w_n)\end{aligned}$$

$$\sum_i \lambda_i = 1$$

Линейная интерполяция (2)

Как подобрать коэффициенты?

- Обучающий корпус – считаем вероятности
- Валидационный корпус – пытаемся максимизировать вероятности входящих в него предложений
- Можем применять полученную модель с коэффициентами на тестовом корпусе!

Сглаживание

Ещё много вариантов

- **Laplace smoothing** (add-one) – добавляем единицу ко всем частотам; вместо 0 всегда 1
- **Kneser-Ney smoothing** – обучаем backoff-модель, чтобы она лучше учитывала зависимости внутри более длинных последовательностей:

San Francisco – частая биграмма, поэтому частоты San и Francisco тоже будут большими.

Но! по отдельности появляться не должны

Генерация текста

- Пример: подсказки в клавиатуре смартфона

Условие — входной текст / префикс:

Пойдем гулять в ... вечером
 вместе
 парк

- LM Sampling – способ тестирования LM

Инструменты

- SRILM

<https://www.sri.com/case-studies/srilm>

- KenLM

<https://kheafield.com/code/kenlm/>

Оценка модели

В целом оценка любой модели NLP:

- **внешняя (extrinsic)**

насколько улучшает качество работы какой-нибудь системы NLP (информационный поиск, машинный перевод, чатботы и т.д.)

- **внутренняя (intrinsic)**

специальные метрики для конкретной задачи

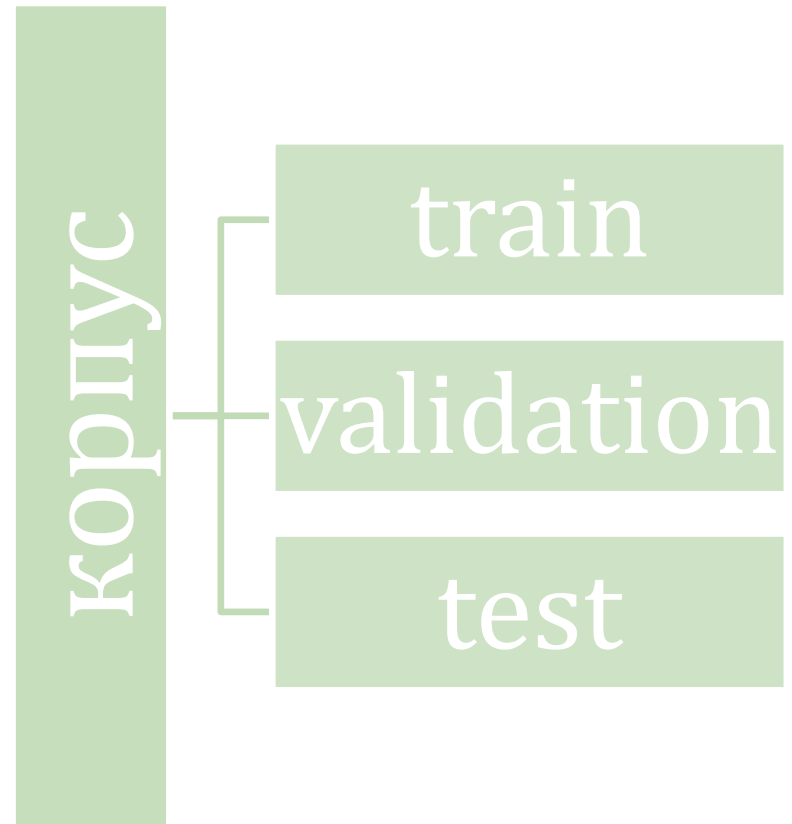
Данные для оценки

Выбор тестовых данных:

- random sampling
- shuffling
- leave-one out strategy

???

Когда перемешивание и выбор случайного сэмпла работают плохо?



Внутренняя оценка языковых моделей

Насколько хорошо модель M оценивает вероятность (не)приемлемых предложений?

> **Энтропия / entropy**

$$H_M(w_1 w_2 \dots w_n) = -\frac{1}{n} \cdot \log P_M(w_1 w_2 \dots w_n)$$

> **Perplexity (перплексия?)** $PPL(M) = 2^{H_M}$

Анализ n-граммных моделей

Достоинства:

- простые, быстро обучаются
- не требуют размеченных данных (возможно, хороший корпус для оценки)

Недостатки:

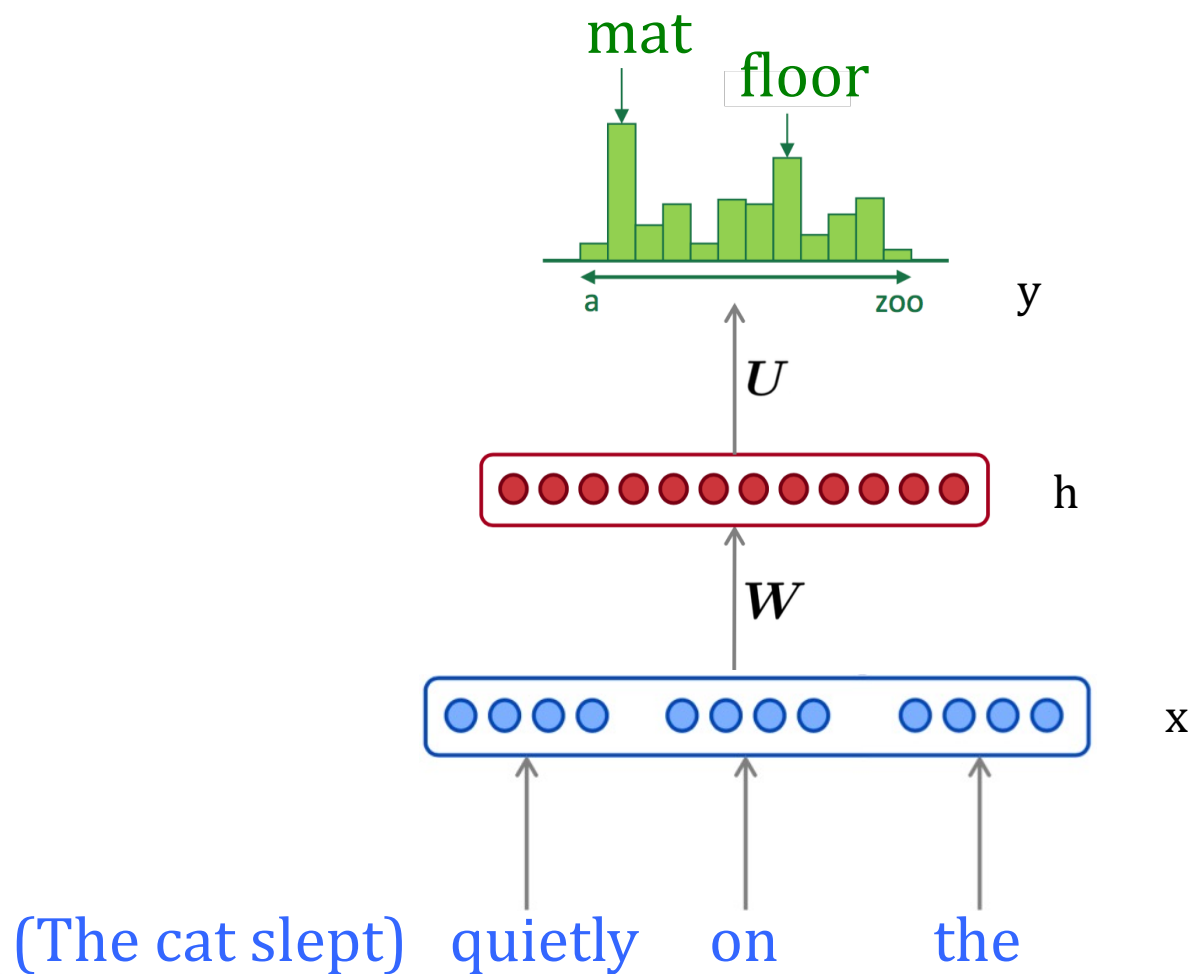
- не моделируют дистантные отношения (согласование, управление, анафора, ... отделяемые приставки и пр.)
- не учитывают морфологию и т.п.
- то есть не обеспечивают **связность (fluency)** текста

Нейронные языковые модели

Neural language models

- Feed-forward LM
- Recurrent neural network (RNN-LM)
- и другие варианты

Нейронные языковые модели



Feed-forward LM

- Объединяем вектора предыдущих k слов

$$x = (x_{i-3}, x_{i-2}, x_{i-1})$$

- Скрытый слой

$$h = f(Wx + b)$$

f – например, логистическая функция (**sigmoid**)

- Выходное распределение

$$z = Uh$$

$$\hat{y} = \text{softmax}(z), \text{ где } \text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^d e^{z_j}}$$

Применение: spell checking

Две задачи:

- error detection — проверка по словарю?
- error correction – выбор похожих слов (расстояние Левенштейна и тп)

Запрос: *погода в казане*

Как использовать тут статистическую LM?

Анонс

- *Ридинг*: будет квиз из вопросов с кратким ответом **20.09**
про word2vec, но заодно и про базовые «кирпичики» нейросетевых моделей
- *Домашка*: подробнее на семинаре
дедлайн **27.09**

Спасибо!

Вопросы?