



О языковых моделях

Евгений Борисов

Языковые модели

языковая модель

предсказываем следующее слово на основе предыдущих
оцениваем правдоподобие цепочки слов

Языковые модели

Приложения

распознавание речи

определение частей речи

генерация текстов

извлечение терминов

поиск и коррекция семантических ошибок

Языковые модели

**корпусы текстов
для обучения языковых моделей**

не размеченные

- Project Gutenberg (<https://www.gutenberg.org>)
- lib.ru

размеченные

- NLTK corpora
- НКРЯ (<http://ruscorpora.ru>)

<https://habr.com/ru/articles/768710/>

Языковые модели

Вероятностные языковые модели

$P(\text{"Дубровский принужден был выйти в отставку"})=?$

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2) \dots P(w_n|w_1^{n-1}) = \prod_{k=1}^n P(w_k|w_1^{k-1})$$

- Предположение Маркова

$$P(w_n|w_1^{n-1}) \approx P(w_n|w_{n-1})$$

- Тогда

$$P(w_1^n) = \prod_{k=1}^n P(w_k|w_{k-1})$$



А. А. Марков

Языковые модели

Оценка вероятностей для слов

метод максимального правдоподобия

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

$C(w)$ — количество слов w в тексте

Оценка цепочки слов (биграммная модель):

$$p(w_1 \dots w_n) = \prod_{k=1}^n p(w_k|w_{k-1})$$

Языковые модели

Оценка цепочки слов (биграммная модель):

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

$$p(w_1 \dots w_n) = \prod_{k=1}^n p(w_k|w_{k-1})$$

проблема: если словарь модели не содержит слова то вероятность биграммы нулевая и оценка цепочки обнуляется

решение: применение методов сглаживания

- **Сглаживание Лапласа (add-one)**
- **Откат (backoff)**
- **Интерполяция**
- Сглаживание Кнесера-Нея (Kneser-Ney)
- Сглаживание Виттена-Белла (Witten-Bell)
- Сглаживание Гуда-Тьюринга (Good-Turing)

Языковые модели

$$p(w_1 \dots w_n) = \prod_{k=1}^n p(w_k | w_{k-1})$$

сглаживание Лапласа

$$p(w_n | w_{n-1}) = \frac{C(w_{n-1} w_n) + 1}{C(w_{n-1}) + V}$$

$C(w)$ – количество слов w в тексте

V – количество слов в словаре модели

достоинства: простая модель, легко реализовать

недостатки: часто показывает неудовлетворительные результаты

Языковые модели

$$p(w_1 \dots w_n) = \prod_{k=1}^n p(w_k | w_{k-1})$$

сглаживание Backoff (откат)

оценка отсутствующих в модели n-gram с помощью k-gram ($0 < k < n$)
т.е. цепочек меньшей длины

$$\hat{P}(w_i | w_{i-2} w_{i-1}) = \begin{cases} \tilde{P}(w_i | w_{i-2} w_{i-1}), C(w_{i-2} w_{i-1} w_i) > 0 \\ \alpha(w_{i-2}^{n-1}) \hat{P}(w_i | w_{i-1}), otherwise \end{cases}$$

Интерполяция

- Смешение вероятностей n-грамм разной длины

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

- при этом $\sum_i \lambda_i = 1$

Языковые модели

Оценка качества языковых моделей

perplexity - коэффициент неопределённости

чем лучше модель предсказывает детали текстовой коллекции тем меньше перплексия

$$\text{perplexity}(P(w_1 \dots w_n)) = \sqrt[n]{\frac{1}{P(w_1 \dots w_n)}}$$

перплексия для биграмной языковой модели

$$\text{perplexity}(P(w_1 \dots w_n)) = \sqrt[n]{\frac{1}{\prod_{k=1}^n P(w_k | w_{k-1})}}$$

Языковые модели

нейросетевая языковая модель (word based model)

- из текстов собираем пары *[[контекст], слово]*
- обучаем RNN по контексту определять слово

input -> LSTM -> softmax

Языковые модели

Литература

git clone https://github.com/mechanoid5/ml_nlp.git

Турдаков Д.Ю. Основы обработки текстов. ИСП РАН, 2017
Лекция 3: Языковые модели и задача определения частей речи.
<https://www.youtube.com/watch?v=seAxPaKw33g>

Анатолий Востряков Языковые модели на все случаи жизни, ODS Data Fest 2018
<https://www.youtube.com/watch?v=TaCbj1kaDQY>