N-граммные модели -2

Языковые модели (language models)

- Определение вероятности предложений, последовательностей слов

- Как вероятна каждая последовательность?
  - P (w1, w2, w3,.. wn)
  - P(w5| w1, w2, w3, w4)

- Языковая модель – математическая модель, которая вычисляется вероятность последовательности слов или условную вероятность следования слова в контексте

Вероятность появления следующего слова

P (Wn|W1,.., Wn-1)= P(W_1,…..Wn)/P(W_1,…Wn-1)

MLE (максимальное правдоподобие)

Pmle (W_1,…Wn)= C(W1,….,Wn)/N

Pmle (Wn|W_1,.., W_{n-1})= C(W_1,…Wn)/ C(W_1,…Wn-1)

C ()- частота появления подстроки

Для биграмм

Pmle (Wn|Wn-1)=C(Wn-1, Wn)/C(Wn-1)

Пример

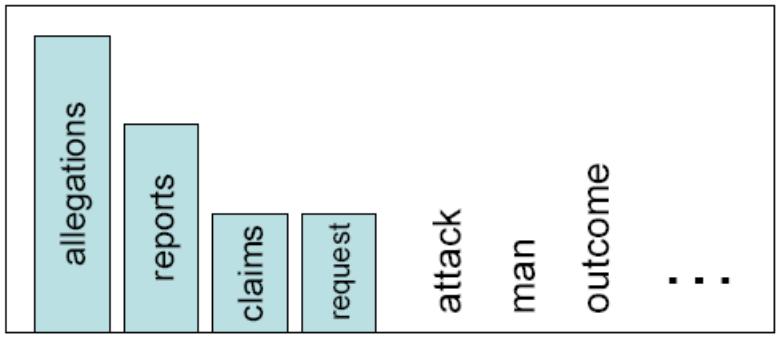
- Корпус
  - <s> Он пошел в школу </s>
  - <s> Пошел он в школу</s>
  - <s> Он не любит мясо</s>

- Вероятности по максимальному праводоподобию?:
  - Униграммы: он, пошел, мясо
  - Биграммы: P(он|пошел), P(пошел|он)

Smoothing is like Robin Hood: Steal from the rich and give to the poor (in probability mass)

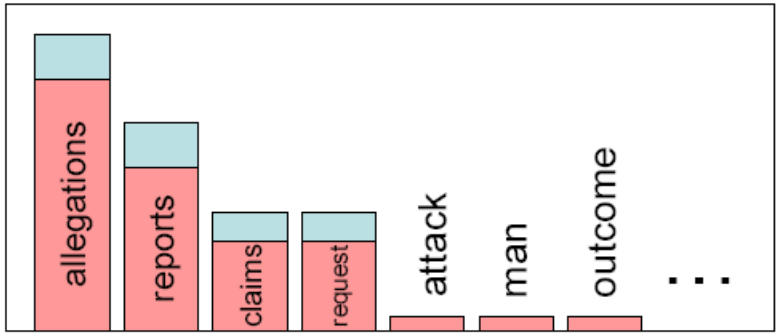
- We often want to make predictions from sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



- Smoothing flattens spiky distributions so they generalize better

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



- Very important all over NLP, but easy to do badly!

Slide from Dan Klein

Сглаживание Лапласа

- Для униграмм:
  - Добавляем 1 к частоте каждого слова
  - Нормализуем N (#tokens) + V (#types)
  - Исходная вероятность униграммы $$P(w_i) = \frac{c_i}{N}$$
  - Новая вероятность униграммы $$P_{LP}(w_i) = \frac{c_i + 1}{N + V}$$
  - Для биграмм
  - исходная $$P(w_n | w_{n-1}) = \frac{c(w_n | w_{n-1})}{c(w_{n-1})}$$
  - новая $$P(w_n | w_{n-1}) = \frac{c(w_n | w_{n-1}) + 1}{c(w_{n-1}) + Vb}$$

Lidstone's Law

$$P_{Lid}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + \lambda}{N + B\lambda}$$

P = вероятность n-граммы

C = частота n-gram в обучающей колллекции

N = количество n-грамм в обучающих данных

B = количество типов (разных n-грамм)

λ : ($0 << \lambda << 1$)

M.L.E: $\lambda = 0$
LaPlace's Law: $\lambda = 1$
Jeffreys-Perks Law: $\lambda = ½$

Пример

- Корпус
  - <s> Он пошел в школу </s>
  - <s> Пошел он в школу</s>
  - <s> Он не любит мясо</s>

- Вероятности по Лапласу?:
  - Униграммы: он, пошел, мясо
  - Биграммы: P(он|пошел), P(пошел|он)

Простая линейная интерполяция (a.k.a., finite mixture models; a.k.a., deleted interpolation)

$$\boldsymbol{P_{li}(w_n \mid w_{n-2}, w_{n-1}) =}$$

$$\lambda_1 P_1(w_n) + \lambda_2 P_2(w_n \mid w_{n-1}) + \lambda_3 P_3(w_n \mid w_{n-2}, w_{n-1})$$

- Взвешенное среднее униграмм, биграмм и триграмм

Пример

- Корпус
  - <s> Он пошел в школу </s>
  - <s> Пошел он в школу</s>
  - <s> Он не любит мясо</s>

- Пусть используется линейная комбинация униграмм, биграмм и триграмм: лямбда=1/3
- Какова вероятность P(в|он, пошел)

Подбор параметров

- Hold out ~ 5 – 10% для тестирования
- Hold out ~ 10% для подбора параметров (smoothing)
- Для тестирования: полезно тестировать на разных коллекциях, и исследовать поведение моделей

How to set the lambdas?

- Use a **held-out** corpus

- Choose λs to maximize the probability of held-out data:
  - Fix the N-gram probabilities (on the training data)
  - Then search for λs that give largest probability to held-out set:

$$\log P(w_1 \ldots w_n \mid M(\lambda_1 \ldots \lambda_k)) = \sum_i \log P_{M(\lambda_1 \ldots \lambda_k)}(w_i \mid w_{i-1})$$

Katz’s Backing-Off

- Используем *n-gram* вероятность, когда достаточно данных
  - (когда частота > k; k usu. = 0 or 1)
- Если нет, то переходим (“back-off”) на *(n-1)-gram* вероятность
- (Повторяем при необходимости)

Идея продвинутых методов сглаживания

- Как оценить вероятность еще не встреченных событий на основе уже увиденных

- **Good-Turing smoothing**
- Kneser-Ney smoothing
- Witten-Bell smoothing

Обозначения

- N_c –количество разных слов с частотностью «с»
- он пошел в школу пошел он в школу он не любит мясо
- $N_3=1$ (он)
- $N_2=3$ (пошел, в , школу)
- $N_1=3$ (не, любит, мясо)

Интуиция

Ловим рыбу: поймали 10 карпов, 3 щуки, 2 белорыбицы, 1 форель, 1 лосося, 1 угря=18 рыб

- Какая вероятность, что следующая рыба – форель? – 1/18
- Какая вероятность, что следующая рыба – новая? – 3/18
- Если принять учет нового, то нужно снизить вероятность встреченного (не 1/18 для лосося)

Пересчет вероятности с помощью Good-Turing

- Для новых $P_{gt}=N_1/N$
- Формула пересчета для других частот

$$c^* = \frac{(c+1)N_{c+1}}{N_c}$$

- где c= count – частота встречаемости
- Для тех, кто не встречался:
- $P_{mle}=0$, $c_{gt0}=N1$, $P_{gt}=N_1/N$ =3/18 (в примере)
- Для тех, кто встречался один раз (в примере)
  - $P_{mle}=1/18$, Cgt1=2*1/3=2/3 Pgt1=1/27 Pgt (Форель)=1/27

Сглаживанеи и реальные частоты (AP Newswire)

R=fmle	flap	fgt	femp
0	0.000137	0.000027	0.000027
1	0.000274	0.446	0.448
2	0.000411	1.26	1.25
3	0.000548	2.24	2.24
4	0.000685	3.24	3.23
5	0.000822	4.22	4.21
6	0.000959	4.19	5.23

Flap – предсказание средней частоты во второй части
по Лапласу
Fgt – предсказание средней частоты во второй части по Good-Turing
Femp – реальная средняя частота во второй части

Как вычислять перплексию

- Перплексия: расчет

$$\begin{aligned} \mathrm{PP}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}} \end{aligned}$$

$$PP = 2^{-\frac{1}{N}\sum_1^N \log_2 p(x)}$$

- Перплексия – это среднее число вариантов, из которых происходит выбор на каждом шаге.
  - Перплексия для предложения, состоящего из случайно последовательности цифр=10
  - Перплексия для униграмм
  - Перплексия для биграмм

Google NGrams

All Our N-gram are Belong to You

By Peter Norvig - 8/03/2006 11:26:00 AM

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects, such as statistical machine translation, speech recognition, spelling correction, entity detection, information extraction, and others. While such models have usually been estimated from training

to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Пример

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensible 40
- serve as the individual 234

Задание к 2 октября

- Выделить десятую часть из Вашего корпуса
- Вычислить вероятности
- Посчитать перплексию на другой части текста (15% общей длины)
  - По униграммам
  - По биграммам
  - В обоих случаях используем закон Jeffreys-Perks Law: $\lambda = ½$
- Отчет
  - Название текста
  - Формулы
  - Необходимые данные
  - Результат вычислений