Natural Language Processing I

5. 平滑处理

WU Xiaokun 吴晓堃

xkun.wu [at] gmail

2022/04/01

平滑处理

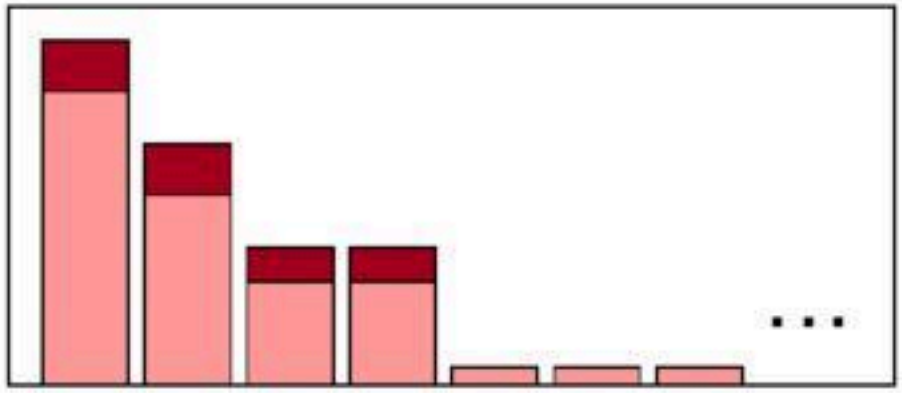
生成未知词汇序列

平滑处理 smoothing/discounting

- 从其他（频率更高的）序列匀一部分**概率质量 probability mass**

原始统计直方图

平滑后的直方图



加一平滑

也称 **Laplace平滑**：灵感来源于（热）扩散过程

- （假装）每个词汇序列看到过比实际值多一次：所有词计数加一

加一平滑

也称 **Laplace平滑**：灵感来源于（热）扩散过程

- （假装）每个词汇序列看到过比实际值多一次：所有词计数加一

例如一元语法：$w_i \in w_{1:N}$，序列$W = w_{1:N}$有N个词

$$P_{MLE}(w_i) = \frac{c_i}{N}$$

$$P_{Add-1}(w_i) = \frac{c_i + 1}{N + |V|}$$

- 注意：分母需要对应增加总共$|V|$个词汇

计数折扣

概率质量被转移了多少?

- 从计数的角度看：数值上打了**折扣**

$$\frac{P_{Add-1}(w_i)}{P_{MLE}(w_i)} = \frac{N}{N+|V|}\frac{(c_i+1)}{c_i}$$
$$= d\frac{c_i+1}{c_i} = \frac{c_i^*}{c_i}$$

- **折扣后的计数**：$c_i^* = \frac{N}{N+|V|}(c_i+1) = d(c_i+1)$;
  - 折扣系数：$d = \frac{N}{N+|V|}$

最大似然估计

回顾：**最大似然估计 Maximum Likelihood Estimation (MLE)**

- 使用训练集**估计模型参数**：得到模型
  - 例如二元语法：二元词汇序列的概率

最大似然估计

回顾：**最大似然估计 Maximum Likelihood Estimation (MLE)**

- 使用训练集**估计模型参数**：得到模型
  - 例如二元语法：二元词汇序列的概率
- 模型训练目标：**最大化训练集的可能性**

例如：1万个词的训练集中，“食堂”出现了100次

- MLE 算得“食堂”的概率是0.01

最大似然估计

回顾：**最大似然估计 Maximum Likelihood Estimation (MLE)**

- 使用训练集**估计模型参数**：得到模型
  - 例如二元语法：二元词汇序列的概率
- 模型训练目标：**最大化训练集的可能性**

例如：1万个词的训练集中，“食堂”出现了100次

- MLE 算得“食堂”的概率是0.01

MLE **深度绑定到已知信息**，估值很粗糙

- 只能说明：在某个1万词的语料库中，“食堂”最有可能出现100次
- 客观实际未必如此：忽略了先验知识
  - 例如：去眼科医院做统计，估计所有人群的疾病分布

实验：二元语法加一平滑

$$P_{MLE}(w_i|w_{i-1}) = \frac{\Gamma(w_{i-1}, w_i)}{\Gamma(w_{i-1})}$$

$$P_{Add-1}(w_i|w_{i-1}) = \frac{\Gamma(w_{i-1}, w_i) + 1}{\Gamma(w_{i-1}) + |V|}$$

注意：（下页）计数值变化剧烈

- 加一平滑本质上是**直接修改统计数据**

实验：计数折扣比较

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

加K平滑

注意：每个词的计数只增加一个小数k（如0.5）

- **减少概率质量的移动**：让扩散过程变慢
- 计数变化会相应减少：降低修改统计数据的影响

$$P_{Add-k}(w_i|w_{i-1}) = \frac{\Gamma(w_{i-1}, w_i) + k}{\Gamma(w_{i-1}) + k|V|}$$

加K平滑

注意：每个词的计数只增加一个小数k（如0.5）

- **减少概率质量的移动**：让扩散过程变慢
- 计数变化会相应减少：降低修改统计数据的影响

$$P_{Add-k}(w_i|w_{i-1}) = \frac{\Gamma(w_{i-1}, w_i) + k}{\Gamma(w_{i-1}) + k|V|}$$

k成为超参数，需要调参

简单修改计数小结

方法过于粗糙，只能作为**备选基准**算法

- 并不适合N元语法

但加一平滑常用于其他NLP模型：**有效解决除0问题**

- 文本分类：关键词更重要
- 数据不太稀疏（0的数量不多）时：计数变化不大、修改不明显

备选、插值

自适应切换方案

回顾：N元模型中的N

- N较大时：对序列建模更好；稀疏性问题更严重
  - 例如$N = 3$：三元词汇序列总数$= 3^{|V|}$

自适应切换方案

回顾：N元模型中的N

- N较大时：对序列建模更好；稀疏性问题更严重
  - 例如$N = 3$：三元词汇序列总数$= 3^{|V|}$

备选法 backoff：自适应切换到尽可能好的方案

- 如果能预测，就使用三元语法
  - 否则试试二元语法；不行还有一元语法

自适应切换方案

回顾：N元模型中的N

- N较大时：对序列建模更好；稀疏性问题更严重
  - 例如$N = 3$：三元词汇序列总数$= 3^{|V|}$

备选法 backoff：自适应切换到尽可能好的方案

- 如果能预测，就使用三元语法
  - 否则试试二元语法；不行还有一元语法
- 类比：在线流媒体播放器**自动切换**分辨率

自适应切换方案

回顾：N元模型中的N

- N较大时：对序列建模更好；稀疏性问题更严重
  - 例如$N = 3$：三元词汇序列总数$= 3^{|V|}$

备选法 backoff：自适应切换到尽可能好的方案

- 如果能预测，就使用三元语法
  - 否则试试二元语法；不行还有一元语法
- 类比：在线流媒体播放器**自动切换**分辨率

插值法 interpolation：取三种模型的加权平均

- 通常比备选法性能更好

线性插值

简单线性插值：固定权重

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

$$\sum_i \lambda_i = 1$$

线性插值

简单线性插值：固定权重

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

$$\sum_i \lambda_i = 1$$

权重λ_i与上下文相关时

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

超参数优化

权重λ_i都是超参数

- 调优参数需要**验证集 validation set**

超参数优化

权重λ_i都是超参数

- 调优参数需要**验证集 validation set**

Expectation Maximization (EM) 算法

1. 固定λ_i，在训练集上**计算**N元语法的概率值
2. 固定N元语法的概率值，在验证集上**优化**λ_i

$$\log P(w_{1:n}|M(\lambda_1..\lambda_K)) = \sum_i \log P_{M(\lambda_1..\lambda_K)}(w_i|w_{i-1})$$

Kneser-Ney 平滑处理

绝对折扣

回顾折扣：将高频词汇的概率质量**部分转移**到未知词汇

- 问题：如何确定计数折扣值?

绝对折扣

回顾折扣：将高频词汇的概率质量**部分转移**到未知词汇

- 问题：如何确定计数折扣值?

[Church 1991] 训练、验证集中**二元词对计数**比较

训练	0	1	2	3	4	5	6	7	8	9
验证	$2.7e^{-5}$	0.448	1.25	2.24	3.23	4.21	5.23	6.21	7.21	8.26

绝对折扣

回顾折扣：将高频词汇的概率质量**部分转移**到未知词汇

- 问题：如何确定计数折扣值?

[Church 1991] 训练、验证集中**二元词对计数**比较

训练	0	1	2	3	4	5	6	7	8	9
验证	$2.7e^{-5}$	0.448	1.25	2.24	3.23	4.21	5.23	6.21	7.21	8.26

$$P_{AbsoluteDiscounting}(w_i|w_{i-1}) = \frac{\Gamma(w_{i-1}, w_i) - d}{\Gamma(w_{i-1})} + \lambda(w_{i-1})P(w_i)$$

根据数据统计：**固定计数折扣**值$d = 0.75$

- 也可在计数为1时，令$d = 0.5$

绝对折扣

回顾折扣：将高频词汇的概率质量**部分转移**到未知词汇

- 问题：如何确定计数折扣值?

[Church 1991] 训练、验证集中**二元词对计数**比较

训练	0	1	2	3	4	5	6	7	8	9
验证	$2.7e^{-5}$	0.448	1.25	2.24	3.23	4.21	5.23	6.21	7.21	8.26

$$P_{AbsoluteDiscounting}(w_i|w_{i-1}) = \frac{\Gamma(w_{i-1}, w_i) - d}{\Gamma(w_{i-1})} + \lambda(w_{i-1})P(w_i)$$

根据数据统计：**固定计数折扣**值$d = 0.75$

- 也可在计数为1时，令$d = 0.5$

思考：折扣后成负数了怎么办?

一元概率

基于**通用词频**的模型性能通常较差，特别是一元语法

他下课后去了_

- “马克思”的词频远高于“食堂”
  - Google Books Ngram Viewer

一元概率

基于**通用词频**的模型性能通常较差，特别是一元语法

他下课后去了_

- “马克思”的词频远高于“食堂”
  - Google Books Ngram Viewer
- 但“马克思”通常跟“主义”**连用**

一元概率

基于**通用词频**的模型性能通常较差，特别是一元语法

他下课后去了_

- “马克思”的词频远高于“食堂”
  - Google Books Ngram Viewer
- 但“马克思”通常跟“主义”**连用**

不应该简单计算“马克思”出现的概率

- 而是计算“马克思”和一个**新词搭配的可能性**

Kneser-Ney

[Kneser and Ney 1995] 词的概率与同时出现的**上下文种类**正相关

- 等价于：二元词汇序列的种类
- 计算：每个二元词汇序列**第一次出现**时，计数加一

$$P_{cont}(w) \propto |v : \Gamma(vw) > 0|$$

Kneser-Ney

[Kneser and Ney 1995] 词的概率与同时出现的**上下文种类**正相关

- 等价于：二元词汇序列的种类
- 计算：每个二元词汇序列**第一次出现**时，计数加一

$$P_{cont}(w) \propto |v : \Gamma(vw) > 0|$$

合理性：如果一个词在很多不同的上下文都出现过，

- 那么这个词也更有可能在新的上下文再次出现。
  - 这个词可以认为是“**胶水词**”：非常容易与其他词连接

Kneser-Ney 概率

首先表述**正相关性**：含v的二元词汇序列的种类

$$P_{cont}(w) \propto |v : \Gamma(vw) > 0|$$

Kneser-Ney 概率

首先表述**正相关性**：含v的二元词汇序列的种类

$$P_{cont}(w) \propto |v : \Gamma(vw) > 0|$$

其次，计算概率需要**归一化**

- 二元词汇序列的总数

$$|(u', w') : \Gamma(u'w') > 0|$$

- 归一化得到概率值

$$P_{cont}(w) = \frac{|v : \Gamma(vw) > 0|}{|(u', w') : \Gamma(u'w') > 0|}$$

Kneser-Ney 插值

绝对折扣与上下文计数的线性组合

$$P_{KN}(w_i|w_{i-1}) = \frac{\max\{\Gamma(w_{i-1}, w_i) - d, 0\}}{\Gamma(w_{i-1})} + \lambda(w_{i-1})P_{cont}(w_i)$$

- 注意分子：折扣成负数时要裁剪为0

Kneser-Ney 插值

绝对折扣与上下文计数的线性组合

$$P_{KN}(w_i|w_{i-1}) = \frac{\max\{\Gamma(w_{i-1}, w_i) - d, 0\}}{\Gamma(w_{i-1})} + \lambda(w_{i-1})P_{cont}(w_i)$$

- 注意分子：折扣成负数时要裁剪为0
- λ 是**正则化常量**

$$\lambda(w_{i-1}) = \frac{d}{\sum_v \Gamma(w_{i-i}v)}|\{w : \Gamma(w_{i-1}w) > 0\}|$$