语言模型的平滑

----有些例子的说明尚不完整，请结合阅读指定参考书的对应部分进行理解

数据平滑

- 给定训练语料S:

BROWN READ HOLY BIBLE

MARK READ A TEXT BOOK

HE READ A BOOK BY DIVID

- 计算句子DAVID READ A BOOK的概率。
- $p(READ|DAVID) = \frac{c(DAVID\ READ)}{\sum_{\omega} c(DAVID\ \omega)} = \frac{0}{1}$

数据平滑

- 必须分配所有可能出现的字符串一个非零的概率值。
- 平滑技术用来解决零概率的问题。
- 平滑处理的基本思想是“劫富济贫”，提高低概率，降低高概率，尽量使概率分布趋于均匀。

数据平滑：加一平滑法 (Laplace smoothing)

- 具体参考《统计自然语言处理》86页。
- 对于二元语法来说，假设每个二元语法出现的次数比实际出现的次数多一次 。
- $p(\omega_i|\omega_{i-1}) = \frac{1+c(\omega_{i-1}\omega_i)}{\sum_{\omega_i}[1+c(\omega_{i-1}\omega_i)]} = \frac{1+c(\omega_{i-1}\omega_i)}{|V|+\sum_{\omega_i} c(\omega_{i-1}\omega_i)}$
- V是考虑的所有词汇的单词表，$|V|$是词汇表单词的个数

数据平滑：加一平滑法

- 给定训练语料S：

BROWN READ HOLY BIBLE

MARK READ A TEXT BOOK

HE READ A BOOK BY DIVID

- $|V| = 11$

数据平滑：加一平滑法

- $p(BROWN\ READ\ A\ BOOK)$

$= p(BROWN| < BOS >) \times p(READ|BROWN) \times p(A|READ)$

$\times p(BOOK|A) \times p(< EOS > |BOOK)$

$= \frac{2}{14} \times \frac{2}{14} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.0001$

数据平滑：加法平滑方法

- 具体参考《统计自然语言处理》87页。
- 对比于加一平滑法，并不假设每个n元语法发生的次数比实际统计次数多一次。
- 假设它比实际出现情况多发生δ次，$0 \leq \delta \leq 1$.
- $p_{add}\left(\omega_i \middle| \omega_{i-n+1}^{i-1}\right) = \frac{\delta + c(\omega_{i-n+1}^{i})}{\delta|V| + \sum_{\omega_i} c(\omega_{i-n+1}^{i})}$
- $\delta = 1$时，有些学者认为这种方法一般表现较差。

数据平滑：Good-Turing估计法

- 具体参考《统计自然语言处理》88页。
- Good-Turing估计法是很多平滑技术的核心。
- 对于任何一个出现r次的n元语法，都假设它出现了r^*次。
- $\mathrm{r}^* = (r+1)\frac{n_{r+1}}{n_r}$
- n_r是训练语料中恰好出现r次的n元语法的数目。

数据平滑：Good-Turing估计法

- 将统计数转换为概率，需进行归一化处理。
- 对于统计数为r的n元语法，其概率公式为：$p_r = \frac{r^*}{N}$
- 其中$N = \sum_{r=0}^{\infty} n_r r^*$

数据平滑：Good-Turing估计法

- $N = \sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1) n_{r+1} = \sum_{r=1}^{\infty} n_r r$
- 因此，N等于这个分布中最初的计数。
- 样本中所有事件的概率之和为$\sum_{r>0} n_r p_r = 1 - \frac{n_1}{N} < 1.$
- $\frac{n_1}{N}$的概率剩余量可以分配给所有未见事件（r=0的事件）

数据平滑：Good-Turing估计法

- Good-Turing方法不能直接用于估计$n_r = 0$的n-gram概率。
- Good-Turing不能实现高阶模型与低阶模型的结合。
- 但Good-Turing方法作为基本方法，会在后续的平滑技术中得到很好的利用。

数据平滑：Katz平滑方法

- 具体参考《统计自然语言处理》89页。
- 当事件在样本中出现的频次大于某一数值k时，运用最大似然估计方法，通过减值估计其概率值；
- 当事件的频次小于k时，使用低阶的语法模型作为代替高阶语法模型的后备，但这种代替受归一化因子的约束。

数据平滑：Katz平滑方法

- 以二元语法模型为例。
- 对于出现次数为$r = c(\omega_{i-1}^{i})$的二元语法ω_{i-1}^{i}，其修正的计数为：
- $$p_{katz}(\omega_{i-1}^{i}) = \begin{cases} d_r \frac{c(\omega_{i-1}^{i})}{c(\omega_{i-1})} & r > 0 \\ \alpha(\omega_{i-1}) p_{ML}(\omega_i) & r = 0 \end{cases}$$

数据平滑：Katz平滑方法

- 所有具有非零计数r的二元语法都根据折扣率d_r被减值了。
- 折扣率d_r近似等于$\frac{r^*}{r}$，由Good-Turing方法预测的。
- 从非零计数中减去的计数量，根据低一阶的分布，被分配给了计数为零的二元语法。
- $p_{ML}(\omega_i)$为ω_i的最大似然估计概率。

数据平滑：Katz平滑方法

- 需选择$\alpha(\omega_{i-1})$的值，使得分布中总的计数$\sum_{\omega_i} c_{katz}(\omega_{i-1}^{i})$保持不变。
- 即$\sum_{\omega_i} c_{katz}(\omega_{i-1}^{i}) = \sum_{\omega_i} c(\omega_{i-1}^{i})$
- $\alpha(\omega_{i-1})$的适当值为

$$\alpha(\omega_{i-1}) = \frac{1 - \sum_{w_i:c(\omega_{i-1}^{i})>0} p_{katz}(\omega_i|\omega_{i-1})}{\sum_{w_i:c(\omega_{i-1}^{i})=0} p_{ML}(\omega_i)} = \frac{1 - \sum_{w_i:c(\omega_{i-1}^{i})>0} p_{katz}(\omega_i|\omega_{i-1})}{1 - \sum_{w_i:c(\omega_{i-1}^{i})>0} p_{ML}(\omega_i)}$$

数据平滑：Katz平滑方法

- 折扣率d_r的计算。
- 大的计数值是可靠的，因此不需要减值。S.M.Katz取$r > k$情况下的$d_r = 1$，并建议k=5.
- $r \leq k$情况下的折扣率，由Good-Turing估计方法计算。其选择遵循如下约束条件：1.最终折扣量与Good-Turing估计预测的减值量成正比。2.全局二元语法分布中被折扣的计数总量等于根据Good-Turing估计应该分配给次数为0的二元语法的总数。

数据平滑：Katz平滑方法

- 第一个约束条件相当于对某些常数μ有公式：

$$1 - d_r = \mu(1 - \frac{r^*}{r})$$

- Good-Turing估计方法预测出应该分配给计数为0的二元语法的计数总量为$n_0 0^* = n_0 \frac{n_1}{n_0} = n_1.$
- 第二个约束条件相当于$\sum_{r=1}^{k} n_r(1 - d_r)r = n_1$

数据平滑：Katz平滑方法

- 由上述约束条件，可得唯一解。

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}$$

数据平滑：Jelinek-Mercer平滑方法

- 具体参考《统计自然语言处理》90页。
- 在一批训练语料上构建二元语法模型，其中两对词的同现次数为0：
  - $c(SEND\ THE) = 0$
  - $c(SEND\ THOU) = 0$
- 则按照加法平滑方法和Good-Turing估计可以得到：
  - $p(THE|SEND) = P(THOU|SEND)$

数据平滑：Jelinek-Mercer平滑方法

- 但直觉上我们认为$p(THE|SEND) > P(THOU|SEND)$，因为THE出现的频率要比THOU高得多。
- 因此可以在二元语法模型中加入一个一元模型。因为一元模型只反映单词的频率。

数据平滑：Jelinek-Mercer平滑方法

- 可以将二元文法模型和一元文法模型进行线性插值。
- $p_{interp}(\omega_i|\omega_{i-1}) = \lambda p_{ML}(\omega_i|\omega_{i-1}) + (1-\lambda)p_{ML}(\omega_i)$
  - 其中$0 \leq \lambda \leq 1.$
- 因为$p_{ML}(THE|SEND) = p_{ML}(THOU|SEND) = 0,$

$p_{ML}(THE) \gg p_{ML}(THOU)$，则$p_{interp}(THE|SEND) >$

$p_{interp}(\mathrm{THOU}|\mathrm{SEND})$

数据平滑：Jelinek-Mercer平滑方法

- 一般来讲，使用低阶的n元模型向高阶插值是有效的，因为当没有足够的语料估计高阶模型的概率时，低阶模型往往可以提供有效信息。
- 通用插值模型：
- $$p_{interp}\left(\omega_i\middle|\omega_{i-n+1}^{i-1}\right) = \lambda_{\omega_{i-n+1}^{i-1}} p_{ML}\left(\omega_i\middle|\omega_{i-n+1}^{i-1}\right) + (1-\lambda_{\omega_{i-n+1}^{i-1}}) p_{ML}\left(\omega_i\middle|\omega_{i-n+2}^{i-1}\right)$$

数据平滑：Jelinek-Mercer平滑方法

- 通用插值模型的含义是：第n阶平滑模型可以递归地定义为n阶最大似然估计模型和n-1阶平滑模型之间的线性插值。
- 递归地结束，可以用最大似然分布作为平滑的1阶模型，或者用均匀分布作为平滑的0阶模型。
- $p_{unif}(\omega_i) = \frac{1}{|V|}$

数据平滑：Jelinek-Mercer平滑方法

- 给定固定的p_{ML}，可以用Baum-Welch算法有效搜索出$\lambda_{\omega_{i-n+1}^{i-1}}$，使某些数据的概率最大。
- 为了得到有意义的结果，估计$\lambda_{\omega_{i-n+1}^{i-1}}$的语料应该与计算$p_{ML}$的语料不同。
- 在留存插值方法(held-out interpolation)中，保留一部分的训练语料来达到这个目的。

数据平滑：Witten-Bell平滑方法

- 具体参考《统计自然语言处理》92页。
- Witten-Bell平滑方法可以认为是Jelinek-Mercer的一个实例。
- N阶平滑模型被递归地定义为n阶最大似然模型和n-1阶平滑模型的线性插值。
- $$p_{WB}\left(\omega_i \middle| \omega_{i-n+1}^{i-1}\right) = \lambda_{\omega_{i-n+1}^{i-1}} p_{ML}\left(\omega_i \middle| \omega_{i-n+1}^{i-1}\right) + (1 - \lambda_{\omega_{i-n+1}^{i-1}}) p_{WB}\left(\omega_i \middle| \omega_{i-n+2}^{i-1}\right)$$

数据平滑：Witten-Bell平滑方法

- 为了计算$\lambda_{\omega_{i-n+1}^{i-1}}$，需要知道$\omega_{i-n+1}^{i-1}$后接的不同单词的数目，将其记作$N_{1+}(\omega_{i-n+1}^{i-1}\cdot)$。
- $N_{1+}\left(\omega_{i-n+1}^{i-1}\cdot\right)=|\{\omega_i: c\left(\omega_{i-n+1}^{i-1}\omega_i\right)>0\}|$
- 其中N_{1+}表示出现过一次或多次的单词的数目，“·”表示统计过程中的自由变量。

数据平滑：Witten-Bell平滑方法

- 可以通过下面的公式定义Witten-Bell平滑参数的$\lambda_{\omega_{i-n+1}^{i-1}}$：
- $1-\lambda_{\omega_{i-n+1}^{i-1}} = \frac{N_{1+}(\omega_{i-n+1}^{i-1}\cdot)}{N_{1+}(\omega_{i-n+1}^{i-1}\cdot)+\sum_{\omega_i} c(\omega_{i-n+1}^{i})}$
- 则$p_{WB}\left(\omega_i \middle| \omega_{i-n+1}^{i-1}\right) = \frac{c\left(\omega_{i-n+1}^{i}\right)+N_{1+}\left(\omega_{i-n+1}^{i-1}\cdot\right)p_{WB}\left(\omega_i \middle| \omega_{i-n+2}^{i-1}\right)}{N_{1+}(\omega_{i-n+1}^{i-1}\cdot)+\sum_{\omega_i} c(\omega_{i-n+1}^{i})}$

数据平滑：绝对值减法

- 具体参考《统计自然语言处理》93页。
- 绝对值减法类似于Jelinek-Mercer平滑方法，涉及高阶和低阶模型的插值问题。
- 绝对值减法通过从每个非零计数中减去一个固定值$D \leq 1$的方法建立高阶分布。

数据平滑：绝对值减法

- $$p_{abs}\left(\omega_i\middle|\omega_{i-n+1}^{i-1}\right)=\frac{\max\{c\left(\omega_{i-n+1}^{i}\right)-D,0\}}{\sum_{\omega_i}c(\omega_{i-n+1}^{i})}+(1-\lambda_{\omega_{i-n+1}^{i-1}})p_{abs}\left(\omega_i\middle|\omega_{i-n+2}^{i-1}\right)$$

- 为了使概率分布之和等于1，取

$$1-\lambda_{\omega_{i-n+1}^{i-1}}=\frac{D}{\sum_{\omega_i}c(\omega_{i-n+1}^{i})}N_{1+}(\omega_{i-n+1}^{i-1}\cdot)$$

数据平滑：绝对值减法

- H. Ney 等人提出了通过训练语料上被删除的估计值来设置D值得方法。
- $D = \frac{n_1}{n_1 + 2n_2}$
- 其中，n_1和n_2是训练语料中分别出现一次和两次的n元语法模型的总数，n是被插值的高阶模型的阶数。

数据平滑：Kneser-Ney平滑方法

- 具体参考《统计自然语言处理》93页。
- 在前面的算法中，通常用平滑后的低阶最大似然分布作为低阶分布。
- 然而，只有当高阶分布中具有极少的或没有计数时，低阶分布在组合模型中才是重要的因素。
- 因此，在这种情况下，应该优化这些参数，来获得较好的性能。

数据平滑：Kneser-Ney平滑方法

- 在一批语料中建立一个二元文法模型。
- 有一个单词FRANCISCO，这个单词只出现在SAN的后面。
- 由于$c(FRANCISCO)$很大，因此一元文法概率$p(FRANCISCO)$也比较大，因此之前的平滑算法就会相应的为出现在新的二元文法历史后面的单词FRANCISCO分配一个高的概率值。
- 然而，这个概率值不应该很高，因为FRANCISCO只跟在唯一的历史后面。

数据平滑：Kneser-Ney平滑方法

- 使用的一元文法的概率不应该与单词出现的次数成比例，而是与它前面的不同单词的数目成比例。
- 只要当前的二元文法没有在前面的语料中出现，一元文法的概率将会是影响当前二元文法概率的较大因素。
- 则要给相应的一元文法分配一个计数，分配给每个一元文法计数的数目就是它前面不同单词的数目。

数据平滑：Kneser-Ney平滑方法

- 在Kneser-Ney平滑方法中，二元文法模型中的一元文法概率就是按照这种方式计算的。
- 但其推导过程为选择的低阶分布必须使得到的高阶平滑分布的边缘概率与训练语料的边缘概率相匹配。

数据平滑：Kneser-Ney平滑方法

- 对于二元文法模型，选择一个平滑的分布p_{KN}，使得对所有的ω_i，满足一元文法边缘概率的约束条件：

$$\sum_{\omega_{i-1}} p_{KN}(\omega_{i-1}\omega_i) = \frac{c(\omega_i)}{\sum_{\omega_i} c(\omega_i)}$$

- 公式左边是平滑的二元文法分布p_{KN}中ω_i的一元文法边缘概率，公式右边是训练语料中ω_i的一元文法频率。

数据平滑：Kneser-Ney平滑方法

- $$p_{KN}\left(\omega_i \middle| \omega_{i-n+1}^{i-1}\right) = \begin{cases} \frac{\max\{\omega_{i-n+1}^{i} - D, 0\}}{\sum_{\omega_i} c(\omega_{i-n+1}^{i})}, & c(\omega_{i-n+1}^{i}) > 0 \\ \gamma_{\omega_{i-n+1}^{i-1}} p_{KN}\left(\omega_i \middle| \omega_{i-n+2}^{i-1}\right), & c(\omega_{i-n+1}^{i}) = 0 \end{cases}$$
- 选择$\gamma_{\omega_{i-n+1}^{i-1}}$使分布之和为1。

数据平滑：后备模型(back-off model)

- 大多数算法可以用下面的公式表示：
- $p_{smooth}(\omega_i|\omega_{i-n+1}^{i-1}) =$

$$\begin{cases} \alpha(\omega_i|\omega_{i-n+1}^{i-1}), c(\omega_{i-n+1}^{i})>0 \\ \gamma_{\omega_{i-n+1}^{i-1}} p_{smooth}(\omega_i|\omega_{i-n+2}^{i-1}), c(\omega_{i-n+1}^{i})=0 \end{cases}$$

数据平滑：后备模型(back-off model)

- 如果n阶语言模型具有非零的计数，就使用$\alpha\left(\omega_i\middle|\omega_{i-n+1}^{i-1}\right)$；
- 否则，回退到低阶分布$p_{smooth}\left(\omega_i\middle|\omega_{i-n+2}^{i-1}\right)$，并选择比例因子$\gamma_{\omega_{i-n+1}^{i-1}}$使得条件概率分布之和等于1.
- 符合这种框架的平滑算法为后备模型。

数据平滑：插值模型 (interpolated model)

- 有些平滑算法采用高阶和低阶n元文法模型的线性插值。
- $$p_{smooth}\left(\omega_i\middle|\omega_{i-n+1}^{i-1}\right)=\lambda_{\omega_{i-n+1}^{i-1}}p_{ML}\left(\omega_i\middle|\omega_{i-n+1}^{i-1}\right)+$$
$$\left(1-\lambda_{\omega_{i-n+1}^{i-1}}\right)p_{smooth}\left(\omega_i\middle|\omega_{i-n+2}^{i-1}\right)$$
- 这种形式的模型称为插值模型。

数据平滑：算法总结

Algorithm	$\alpha(w_i \mid w_{i-n+1}^{i-1})$	$\gamma(w_{i-n+1}^{i-1})$	$p_{\text{smooth}}(w_i \mid w_{i-n+2}^{i-1})$
Additive	$\frac{c(w_{i-n+1}^{i})+\delta}{\sum_{w_i} c(w_{i-n+1}^{i})+\delta \mid V \mid}$	0	n. a.
Jelinek-Mercer	$\lambda_{w_{i-n+1}^{i-1}} p_{\text{ML}}(\ )+\cdots$	$(1-\lambda_{w_{i-n+1}^{i-1}})$	$p_{\text{interp}}(w_i \mid w_{i-n+2}^{i-1})$
Katz	$\frac{d_r r}{\sum_{w_i} c(w_{i-n+1}^{i})}$	$\frac{1-\sum_{w_i : c(w_{i-n+1}^{i})>0} p_{\text{Katz}}(w_i \mid w_{i-n+1}^{i-1})}{\sum_{w_i : c(w_{i-n+1}^{i})=0} p_{\text{Katz}}(w_i \mid w_{i-n+2}^{i-1})}$	$p_{\text{Katz}}(w_i \mid w_{i-n+2}^{i-1})$
Witten-Bell	$(1-\gamma(w_{i-n+1}^{i-1}))\, p_{\text{ML}}(\ )+\cdots$	$\frac{N_{1+}(w_{i-n+1}^{i-1}\ \cdot)}{N_{1+}(w_{i-n+1}^{i-1}\ \cdot)+\sum_{w_i} c(w_{i-n+1}^{i})}$	$p_{\text{WB}}(w_i \mid w_{i-n+2}^{i-1})$
Absolute disc.	$\frac{\max\{c(w_{i-n+1}^{i})-D,0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}+\cdots$	$\frac{D}{\sum_{w_i} c(w_{i-n+1}^{i})} N_{1+}(w_{i-n+1}^{i-1}\ \cdot)$	$p_{\text{abs}}(w_i \mid w_{i-n+2}^{i-1})$
Kneser-Ney (interpolated)	$\frac{\max\{c(w_{i-n+1}^{i})-D,0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}+\cdots$	$\frac{D}{\sum_{w_i} c(w_{i-n+1}^{i})} N_{1+}(w_{i-n+1}^{i-1}\ \cdot)$	$\frac{N_{1+}(\cdot\ w_{i-n+2}^{i})}{N_{1+}(\cdot\ w_{i-n+2}^{i-1}\ \cdot)}$

数据平滑：算法总结

- 后备模型和插值模型的根本区别在于，在确定非零计数的n元文法的概率时，插值模型使用低阶分布的信息，而后备模型不使用。
- 但不管是后备模型还是插值模型，都使用了低阶分布来确定计数为零的n元语法的概率。

数据平滑：平滑方法的比较

- 影响最大的因素是采用修正的后备分布，例如Kneser-Ney平滑方法采用的后备分布。这可能是Kneser-Ney优于其他方法的基本原因。
- 绝对减值优于线性减值。对于较低的计数来说，理想的平均减值上升很快，而对于较大的计数，则变得比较平缓。

数据平滑：平滑方法的比较

- 从性能上看，对于较低的非零计数，插值模型大大的优于后备模型，这是因为低阶模型在为较低计数的n元语法确定恰当的减值时提供了优化这些有价值的信息。
- 增加算法的自由参数，并在留存数据上优化这些参数，可以改进算法的性能。