第5章 语言模型

基本概念

基于大规模语料库和统计方法，可以 {发现语言使用的普遍规律 / 进行机器学习，自动获取语言知识 / 对未知语言现象进行预测}

计算一段文字(如句子)的概率　(以句子为单位 × $p(w_1) \times p(w_2) \cdots p(w_n)$ ×)

语句 $s = w_1 w_2 \cdots w_m$ 的先验概率：

$$p(s) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_1 w_2) \times \cdots \times p(w_m|w_1 \cdots w_{m-1})$$

$$= \prod_{i=1} p(w_i|w_1 \cdots w_{i-1})$$

当 $i=1$ 时，$p(w_1|w_0) = p(w_1)$

当 w_i 可以是字、词、短语或词类等，称为统计基元

注：(1) w_i 可以是字、词、短语或词类等，称为统计基元

(2) 由 $w_1, \cdots, w_{i-1}$ 构成的一个序列，称为 w_i 的历史

问题：自由参数的个数可达 L^m

解决方法：减少历史基元的个数，即将 $w_1, w_2, \cdots w_{i-1}$ 映射到等价类 $s(w_1, w_2, \cdots, w_{i-1})$，使等价类的数目远小于原来不同历史基元的数目

则有：$p(w_i|w_1, \cdots, w_{i-1}) = p(w_i|s(w_1, \cdots, w_{i-1}))$

等价类的划分：假设只与前 $n-1$ 个基元有关

即：n元文法模型 (n-gram)
- $n=1$　一元文法　独立于历史　unigram　monogram
- $n=2$　1阶马尔可夫链　bigram/bi-gram, 二元文法
- $n=3$　2阶马尔可夫链　trigram/tri-gram …

为保证条件概率在 $i=1$ 时有意义，同时为了保证句子内所有字符串的概率和为1，则：

<BOS> $w_1 w_2 \cdots w_m$ <EOS>

不失一般性，对于 $n>2$ 的 n-gram，$p(s)$ 可以分解为：

$$p(s) = \prod_{i=1}^{m+1} p(w_i|w_{i-n+1}^{i-1})$$

其中 w_i^j 表示词序列 $w_i \cdots w_j$，w_{i-n+1} 从 w_0 开始，w_0 为 <BOS>，w_{m+1} 为 <EOS>

例：<BOS> John read a book <EOS>

基于2元文法的概率为：

$$p(\text{John read a book}) = p(\text{John}|\langle BOS\rangle) \times p(\text{read}|\text{John}) \times p(\text{a}|\text{read}) \times p(\text{book}|\text{a}) \times p(\langle EOS\rangle|\text{book})$$

应用：1. 音节转换问题

$$\hat{C}string = \arg\max_{Cstring} P(Cstring|Pinyin)$$

$$= \arg\max_{Cstring} \frac{P(Pinyin|Cstring)\times P(Cstring)}{P(Pinyin)}$$

$$= \arg\max P(Cstring)$$

Cstring的概率即可使用 2-gram计算

一元语法 样本空间为N，二元为N^2 ...

汉字：四元模型

应用实例：微软拼音输入法.

2. 汉语分词问题

5.2 参数估计

训练语料 (training data)：

最大似然估计 (maximum likelihood evaluation, MLE)：相对频率计算概率

对于n-gram，参数 $P(w_i|w_{i-n+1}^{i-1})$ 可由最大似然估计求得

$$P(w_i|w_{i-n+1}^{i-1}) = f(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i})}$$

例：training data: "John read Moby Dick"

...

...

$P(John|\langle BOS\rangle) = ..$

数据平滑：解决数据匮乏(稀疏)(sparse data)引起的零概率问题

data smoothing

基本目标：测试样本的语言模型困惑度越小越好

基本约束：$\sum_{w_i} P(w_i|w_1, w_2, \dots, w_{i-1}) = 1$

对于一个平滑的n-gram，其概率为 $P(w_i|w_{i-n+1}^{i-1})$，可以计算句子的概率：

$$P(s) = \prod_{i=1}^{m+1} P(w_i|w_{i-n+1}^{i-1})$$

假设测试语料T由l_T个句子构成$(t_1, \dots, t_{l_T})$，则整个测试集的概率为

$$P(T) = \prod_{i=1}^{l_T} P(t_i)$$

模型 $p(w_i/w_{i-n+1}^{i-1})$ 对于测试语料的交叉熵：

$$H_p(T) = -\frac{1}{W_T}\log_2 p(T)$$

其中，W_T 是以词为文本 T 的词数

模型 p 的困惑度 $PP_p(T)$ 定义为：$PP_p(T) = 2^{H_p(T)}$

n-gram 对于英语文本的困惑度：
50 ~ 1000
交叉熵：6 ~ 10 bits/word

数据平滑方法：

加1法：$\frac{1}{5}$ 0 $\frac{2}{5}$

Additive smoothing $\frac{2}{6}$ $\frac{1}{6}$ $\frac{3}{6}$

对于2-gram: $p(w_i/w_{i-1}) = \frac{1+C(w_{i-1}w_i)}{\sum_{w_i}[1+C(w_{i-1}w_i)]}$

减值法/折扣法（Discounting）

修改 training data 中的实际计数，使原来的概率和小于1，其余部分

① Good-Turing 估计

n_r（样本中正好出现 r 次的事件的数目）

$N = \sum_{r=1}^{\infty} n_r r$

$= \sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1) n_{r+1}$　　r^* 是比 r 的小

即 $r^* = (r+1)\frac{n_{r+1}}{n_r}$

那么，Good-Turing 估计在样本中出现 r 次的事件的概率为：$P_r = \frac{r^*}{N}$

适用于：大词汇集产生的符合多项式分布的大量的观察数据

归一化 $\hat{P}_r = \frac{P_r}{\sum_r P_r}$

② Back-off 法（后备/后退）法　Katz 后退法

对非0事件按 Good-Turing 法计算减值，节省出来的概率按低阶分布分给

0概率事件

$$P(\text{Katz}(w_i/w_{i-1})) = \begin{cases} d_r \frac{C(w_{i-1}w_i)}{C(w_{i-1})} & \text{若 } C(w_{i-1}w_i) = r > 0 \\ \alpha(w_{i-1}) P_{ML}(w_i) & \text{若 } C(w_{i-1}w_i) = 0 \end{cases}$$

d_r（$0 < d_r < 1$）折扣率　$P_{ML}(w_i)$：w_i 的最大似然估计

$\approx r^*/r$

$\alpha(w_{i-1})$的确定：

$$\sum_{w_3} P_{katz}(w_3|w_{i-1}) = 1$$

$$\sum_{w_i: r=0} \alpha(w_{i-1}) P_{ML}(w_i) + \sum_{w_i: r>0} P_{katz}(w_i|w_{i-1}) = 1$$

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w_i: r>0} P_{katz}(w_i|w_{i-1})}{\sum_{w_i: r=0} P_{ML}(w_i)}$$

③ 绝对减值法（Absolute discounting）　　适用于2元及以上模型

从每个计数r中减去同样的量，针对语料事件。

$$P_r = \begin{cases} \frac{r-b}{N} & \text{若 } r>0 \\ \frac{b(R-n_0)}{Nn_0} & \text{若 } r=0 \end{cases}$$

n_0：未出现事件的数目　　b：b减成去的常量

$\frac{b(R-n_0)}{N}$：由于减值而产生的剩余概率

④ 线性减值法.

$$P_r = \begin{cases} \frac{c(w)r}{N} & \text{若 } r>0 \\ \frac{\alpha}{n_0} & \text{若 } r=0. \end{cases}$$

$\alpha = \frac{n_1}{N}$　　表从每个计数r中减去与该个数成正比例量

绝对减值优于线性减值

3). 删除插值法.

用低阶语法为什高阶语法　　3-gram → 4-gram
2-gram → 3-gram
1-gram → 2-gram.

$$P(w_3|w_1w_2) = \lambda_3 P'(w_3|w_1w_2) + \lambda_2 P'(w_3|w_2) + \lambda_1 P'(w_3)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

$\lambda_1, \lambda_2, \lambda_3$的确定：将训练语料分为2部分，即从原始中删一部分作为留存数据，第一部分用于

估计 $p'(w_3|w_1w_2)$, $p'(w_3|w_2)$ 和 $p'(w_3)$

第二部分用于计算 λ_1, λ_2, λ_3, 使理论估计小

5.4 语言模型的自适应

问题: ①. 综合性语料难以反映不同领域之间在语言使用规律上的差异. 而语言模型训练对训练文本的类型、主题和风格等都十分敏感

②. n元语言模型的局部性假设在很多情况下都明显不成立

自适应方法:

① 基于缓存的语言模型 (cache-based LM)

② 基于混合方法的语言模型

③ 基于最大熵的语言模型

①. 在文本中刚出现的词在后边可能再次出现的可能性往往较大

线性插值: $p(w_i|w_1^{i-1}) = \lambda p_{cache}(w_i|w_1^{i-1}) + (1-\lambda) p_{n\text{-}gram}(w_i|w_{i-n+1}^{i-1})$

λ 可通过 EM 算法求得. cache: 1000 ~ 3000 个

$$\hat{p}_{cache}(w_i|w_1^{i-1}) = \frac{1}{k}\sum_{j=i-k}^{i-1} I\{w_j = w_i\}$$

I: 指示函数, 如果 ξ 表示的情况出现, 则 $I_\xi = 1$, 否则 $I_\xi = 0$

缺陷: 缓存里的词对当前词的影响是一样的

改进: $\hat{p}_{cache}(w_i|w_1^{i-1}) = \beta \sum_{j=1}^{i-1} I_{\{w_i = w_j\}} e^{-\alpha(i-j)}$

α: 衰减率 β: 归一化参数 $\sum_{w\in V} \hat{p}_{cache}(w_i|w_1^{i-1}) = 1$

②. 来源不同, 识别同源 (homogeneous)

划分为不同 n 个子模型 $m_1, \cdots m_n$ (聚类)

$$\hat{p}(w_i|w_1^{i-1}) = \sum_{j=1}^{n} \lambda_j p_{m_j}(w_i|w_1^{i-1})$$

其中, $0 \le \lambda_j \le 1$, $\sum_{j=1}^{n} \lambda_j = 1$

λ 值通过 EM 算法计算

EM 迭代计算插值系数 λ：

a) 对于 n 个类，随机初始化插值参数 λ

b) 计算新的隐半物期望

c) $\lambda_j^{(i+1)} = \frac{\lambda_j^{(i)} P_j(w|h)}{\sum_{j=1}^{n} \lambda_j^{(i)} P_j(w|h)}$ 为历史

d) 迭代直至收敛

理论先验，实际一般为线

⑤. 两个语言模型 M_1 和 M_2

$$P_{M_1}(w_i | w_1^{i-1}) = f(w_i, w_{i-1})$$

M_2 是距离为 2 的 2-gram 模型 $P_{M_2}(w_i | w_1^{i-1}) = g(w_i, w_{i-2})$

M_1 是标准 2-gram 模型

线性插值取这两个概率估计的平均值，使用 EM 算法选择使困惑度最小的模型。

5.5 语言模型应用举例

1. 汉语分词 $\hat{w} \approx \arg\max_w p(w)$

其中 $s = z_1 z_2 \cdots z_m$，$w = w_1 w_2 \cdots w_k$ 是一种可能的切分。

具体实现时，可把汉语词汇分成如下几类：

(1) 词典中

(2) 词典中未登录但能生成的词

(3) 与数字相关的实体

(4) 专用名词

把一个可能的词序列 w 转换成词类序列 $C = c_1 c_2 \cdots c_n$

① 人名 PN，地名 LN，机构名 ON

② 日期 date，时间 time，百分比 per，货币 mon

③ 词典中词 IW，词典外词 UW，每个词单独一类

将词序列变成类序列

$$C = \arg\max P(C|S)$$
$$= \arg\max P(C) \times P(S|C)$$

↑ 语言模型　　↑ 生成模型

$$P(C) = P(c_1) \times P(c_2|c_1) \times \prod_{i} P(c_i|c_{i-2}c_{i-1})$$

极大似然估计

$$P(S|C) \approx \prod_{i=1}^{n} P(s_i|c_i)$$

近似假设

例如：如果"教授"是词表中的词，那么 $P(s_i=\text{教授}|c_i=LW)=1$，否则 $P(s_i|c_i)=0$

生成模型则为判断词类类别

2. 分词与词性标注一体化系统

句S：$W = w_1 w_2 \cdots w_n$

单词 $w_i (1 \le i \le n)$ 的词性标注为 t_i，$T = t_1 t_2 \cdots t_n$

任务：在S所对应的各种切分和标注形式中，寻找最优的 $P(W,T)$

(1). 基于词性的三元统计模型

$$P(W,T) = P(W|T) \times P(T) \approx \prod_{i} P(w_i|t_i) \times P(t_i|t_{i-1}, t_{i-2}) \cdots$$

$P(W|T)$ 为生成模型，$P(T)$ 为基于词性的语言模型。

(2). 基于单词的三元统计模型

$$P(W,T) = P(T|W) \times P(W) \approx \prod_{i} P(t_i|w_i) \times P(w_i|w_{i-1}, w_{i-2})$$

(3). 分词与词性标注一体化模型

$$P^*(W,T) = \alpha \prod_{i} P(w_i|t_i) \times P(t_i|t_{i-1}, t_{i-2}) + \beta \prod_{i} P(t_i|w_i) \times P(w_i|w_{i-1}, w_{i-2})$$

指导思想：通过调整系数 α 和 β 的值来确定两个子模型在整个分词与词性标注中所发挥作用的比重。

分析公式(2)

$p(t_i|w)$ 对当词无帮助，但在为词确也后对词性标注会增添偏差

因此删掉，并令 $\alpha=1$.

$$p(w,T)=\prod_{i=1}^{n} p(w_i|t_i)\times p(t_i|t_{i-1},t_{i-2})+\beta\vec{\sum} p(w_i|w_{i-1},w_{i-2})$$

确定 β 的值：$\beta=\frac{\text{词典中词 } w \text{ 的个数}}{\text{词性 } t \text{ 的种类数}}$