第5章 语言模型

(1/3)

语言模型

语言模型本质上是在回答一个问题：出现的语句是否合理。

如：

$$P(I\ am\ Light) > P(Light\ I\ am)$$

通俗解释：判断一个语言序列是否是正常语句，即是否是**人话**。

好的语言模型,对合理的语句会给出较高的概率，反之，会给出极低的概率。

语言模型广泛应用于语音识别、手写体文字识别、机器翻译、键盘输入、信息检索等领域。

本章内容

- 5.1 传统语言模型
- 5.2 神经语言模型
- 5.3 文本表示

5.1 传统语言模型

- 5.1.1 n元文法
- 5.1.2 参数估计
- 5.1.3 数据平滑方法
- 5.1.4 语言模型自适应
- 5.1.5 n元文法模型的应用

5.1.1 n元文法

如何计算一段文字(句子)的概率？

阳春三月春意盎然，少先队员脸上荡漾着喜悦的笑容，鲜艳的红领巾在他们的胸前迎风飘扬。

- 以一段文字(句子)为单位统计相对频率？
- 根据句子构成单位的概率计算联合概率?

$$p(w_1)\times p(w_2)\times\cdots\times p(w_n)$$

5.1.1 n元文法

语句 $s = w_1\ w_2\ \ldots\ w_m$ 的先验概率：

$$p(s) = p(w_1)\times p(w_2|w_1)\times p(w_3|w_1w_2)\times\ldots\times p(w_m|w_1\ldots w_{m\text{-}1})$$

$$= \prod_{i=1}^{m} p(w_i \mid w_1 \mathrm{L}\ \ w_{i-1}) \qquad \ldots (5\text{-}1)$$

当 i=1 时，$p(w_1|w_0) = p(w_1)$。

说明：(1) w_i 可以是字、词、短语或词类等，统称为统计基元。(2) w_i 的概率取决于 $w_1, \ldots, w_{i-1}$，条件序列 $w_1, \ldots, w_{i-1}$ 称为 w_i 的历史(history)。

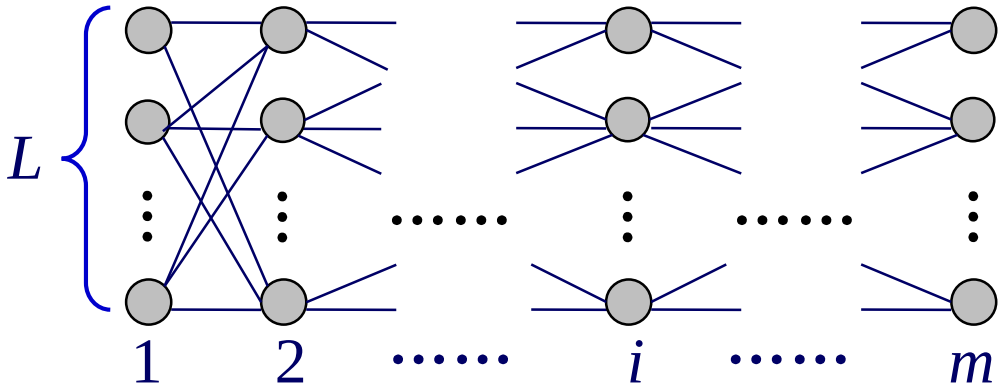
5.1.1 n元文法

$$
\begin{aligned}
& p(\text{我是一个学生}) \\
= \ & p(\text{我}, \text{是}, \text{一}, \text{个}, \text{学生}) \\
= \ & p(\text{我}) \cdot \\
& p(\text{是} \mid \text{我}) \cdot \\
& p(\text{一} \mid \text{我}, \text{是}) \cdot \\
& p(\text{个} \mid \text{我}, \text{是}, \text{一}) \cdot \\
& p(\text{学生} \mid \text{我}, \text{是}, \text{一}, \text{个})
\end{aligned}
$$

5.1.1 n元文法

问题：随着历史基元数量的增加，不同的“历史”组合构成的路径数量指数级增长。对于长度为m的句子，模型中有L^m 个自由参数 $p(w_m|w_1 \ldots w_{m-1})$。$L$为词汇表中基元(如词)的数量。

如果 L=6763, m=3, 自由参数的数目为 3.09×10^{11}！



5.1.1 n元文法

- 问题的解决方法

由于数据稀疏和系统处理能力的限制，统计语言建模只考虑有限长度的历史。

通过将语言模拟成$N-1$阶马尔科夫链，n元文法（N-gram)模型减少了参数估计的维数：

$$p(w_i|h_i) \approx p(w_i|w_{i-N+1}, \cdots, w_{i-1})$$

即假设当前词出现的概率只依赖于前N−1个词。

5.1.1 n元文法

- 当 ***n*=1** 时，即出现在第 ***i*** 位上的词 ***wi*** 独立于历史。一元文法也被写为 uni-gram 或 monogram；
- 当 ***n*=2** 时, **2-gram (**bi-gram**)** 被称为**1**阶马尔可夫链；
- 当 ***n*=3** 时, **3-gram(**tri-gram**)**被称为**2**阶马尔可夫链，

依次类推。

5.1.1 n元文法

例如：

对于垃圾邮件中的语句s=“我/司/可/办理/正规/发票/保真/增值税/发票/点数/优惠”，使用bi-gram模型:

P(s)=P(“ 我 ”)P(“ 司 ”|“ 我 ”)P(“ 可 ” |“ 司 ”)P(“ 办理”|“可”)...P(“优惠” |“点数”)

如果使用tri-gram：

P(s)=P(“我”)P(“司” |“我”)P(“可” |“我” ,“司”) P(“办理” |“司” ,“可”) ...P(“优惠” |“发票” ,“点数”)

5.1.1 n元文法

为了保证条件概率在i=1时有意义，同时保证句子内所有字符串的概率和为 1，即$\sum_s p(s)=1$，可以在句子首尾两端增加两个标志: <BOS> w_1 w_2 … w_m <EOS>。不失一般性，对于n>2的 n-grams，$p(s)$ 可以分解为：

$$p(s)=\prod_{i=1}^{m+1} p(w_i \mid w_{i-n+1}^{i-1}) \qquad \text{… (5-4)}$$

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

5.1.1 n元文法

- 应用示例：音字转换问题

输入拼音串：ta shi yan jiu sheng wu de　　*Py*

可能的汉字：

CStri
- 踏实研究生物的
- 他实验救生物的
- 他使烟酒生物的
- 他是研究生物的
- ……

$$CStri = \arg\max_{CStri} p(CStri \mid Py)$$

$$= \arg\max_{CStri} \frac{p(Py \mid CStri) \times p(CStri)}{p(Py)}$$

$$= \arg\max_{CStri} p(Py \mid CStri) \times p(CStri)$$

$$\approx \arg\max_{CStri} p(CStri)$$

5.1.1 n元文法

CStri ={踏实 研究 生物 的, 他 实验 救生 物 的,
他 是 研究生 物 的, 他 使 烟 酒 生 雾 的, ……}

如果使用 2-gram：

$p(CStri_1)$ =p(踏实|<BOS>) × p(研究|踏实) × p(生物|研究) ×
p(的|生物) × p(<EOS>|的)

$p(CStri_2)$ =p(他|<BOS>) × p(实验|他) × p(救生|实验) ×
p(物|救生) × p(的|物) × p(<EOS>|的)

……

对于汉字而言，4元文法效果会好一些。智能狂拼、微软拼音输入法都是基于*n*-gram 实现的。

问题：如何获得 ***n*** 元文法模型？

5.1 传统语言模型

5.1.1 n元文法

5.1.2 参数估计

5.1.3 数据平滑方法

5.1.4 语言模型自适应

5.1.5 n元文法模型的应用

5.1.2 参数估计

- 基本思路
  - 收集、标注大规模样本，我们称其为训练数据/语料(*training data / corpus*)。
  - 利用最大似然估计(*maximum likelihood evaluation*, MLE)方法计算概率。

5.1.2 参数估计

对于 n-gram，参数 $p(w_i \mid w_{i-n+1}^{i-1})$ 通过最大似然估计计算:

$$p(w_i \mid w_{i-n+1}^{i-1}) = f(w_i \mid w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i})} \quad \ldots(5\text{-}5)$$

其中，$\sum_{w_i} c(w_{i-n+1}^{i})$ 是历史串 w_{i-n+1}^{i-1} 在给定语料中出现的次数。$f(w_i \mid w_{i-n+1}^{i-1})$ 是在给定 w_{i-n+1}^{i-1} 的条件下 w_i 出现的相对频度，分子为 w_{i-n+1}^{i-1} 与 w_i 同现的次数。

5.1.2 参数估计

例如，给定训练语料：

John read Moby Dick，

Mary read a different book,

She read a book by Cher

根据 2 元文法求句子*John read a book*的概率？

$$p(John|<BOS>)=\frac{c(<BOS>John)}{\sum_w c(<BOS>w)}=\frac{1}{3}$$

$$p(read\mid John)=\frac{c(John\quad read)}{\sum_w c(John\quad w)}=\frac{1}{1}$$

5.1.2 参数估计

$$p(a \mid read) = \frac{c(read \quad a)}{\sum_w c(read \quad w)} = \frac{2}{3} \qquad p(book \mid a) = \frac{c(a \quad book)}{\sum_w c(a \quad w)} = \frac{1}{2}$$

$$p(< EOS > \mid book) = \frac{c(book \quad < EOS >)}{\sum_w c(book \quad w)} = \frac{1}{2}$$

$$p(John\ read\ a\ book) = \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$$

<BOS>John read Moby Dick<EOS>
<BOS>Mary read a different book<EOS>
<BOS>She read a book by Cher<EOS>

5.1.2 参数估计

$$p(Cher\ read\ a\ book) = p(Cher|<BOS>) \times p(read|Cher) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

新的待测语句

$$p(Cher|<BOS>) = \frac{c(<BOS>\ Cher)}{\sum_w c(<BOS>\ w)} = \frac{0}{3}$$

$$p(read|Cher) = \frac{c(Cher\ \ read)}{\sum_w c(Cher\ \ w)} = \frac{0}{1}$$

于是，$p(Cher\ read\ a\ book) = 0$

数据匮乏/稀疏
(sparse data)

数据平滑
(data smoothing)

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

5.1 传统语言模型

5.1.1 n元文法

5.1.2 参数估计

5.1.3 数据平滑方法

5.1.4 语言模型自适应

5.1.5 n元文法模型的应用

5.1.3 数据平滑方法

◆基本思想：

调整最大似然估计的概率值，使零概率增值，使非零概率下调，利用“劫富济贫”，消除零概率，改进模型的整体正确率。

- 平滑方法目标：测试样本的语言模型困惑度越小越好。
- 约束：$\sum_{w_i} p(w_i \mid w_{i-n+1}^{i-1}) = 1$

5.1.3 数据平滑方法

- 回顾—困惑度：

假定平滑的n-gram概率为 $p(w_i \mid w_{i-n+1}^{i-1})$ ，句子s 的概率：

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

假定测试语料T 由 l_T 个句子构成：$(s_1, s_2, \mathrm{L}\ , s_{l_T})$，共含 w_T 个词，那么，整个测试集的概率为：$p(T) = \prod_{i=1}^{l_T} p(s_i)$

交叉熵：$H_p(T) = -\frac{1}{w_T} \log_2 p(T)$　　困惑度：$PP_P(T) = 2^{H_p(T)}$

5.1.3 数据平滑方法

- 回顾—困惑度：

模型好坏与N值和 $p(w_i \mid w_{i-n+1}^{i-1})$ 相关

- Results from Goodman ("A bit of progress in language modeling"), where $|\mathcal{V}| = 50,000$
- A trigram model: $p(x_1 \ldots x_n) = \prod_{i=1}^{n} q(x_i | x_{i-2}, x_{i-1})$. Perplexity $= 74$
- A bigram model: $p(x_1 \ldots x_n) = \prod_{i=1}^{n} q(x_i | x_{i-1})$. Perplexity $= 137$
- A unigram model: $p(x_1 \ldots x_n) = \prod_{i=1}^{n} q(x_i)$. Perplexity $= 955$

5.1.3 数据平滑方法

◆数据平滑方法

(1)加1法 (additive)

(2)减值法/折扣法 (discounting)

(3)删除插值法 (deleted interpolation)

5.1.3 数据平滑方法

(1) 加1法(Additive smoothing)

也称拉普拉斯平滑。

基本思想:为了避免零概率问题，将N-gram模型中每个N元对的出现次数加1。

对于2-gram 有：

$$p(w_i \mid w_{i-1}) = \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i}[1 + c(w_{i-1}w_i)]}$$

V 为被考虑语料的词汇表。

$$= \frac{1 + c(w_{i-1}w_i)}{|V| + \sum_{w_i} c(w_{i-1}w_i)}$$

5.1.3 数据平滑方法

在前面 3 条训练样本的例子中,

$$p(Cher\ read\ a\ book) = p(Cher|<BOS>) \times p(read|Cher) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

原来:

$p(Cher|<BOS>) = 0/3$

$p(read|Cher) = 0/1$

$p(a|read) = 2/3$

$p(book|a) = 1/2$

$p(<EOS>|book) = 1/2$

5.1.3 数据平滑方法

词汇量：$|V|=11$

原来：

$p(Cher|<BOS>) = 0/3$

$p(read|Cher) = 0/1$

$p(a|read) = 2/3$

$p(book|a) = 1/2$

$p(<EOS>|book)= 1/2$

平滑以后：

$p(Cher|<BOS>) = (0+1)/(11+3) = 1/14$

$p(read|Cher) = (0+1)/(11+1) = 1/12$

$p(a|read) = (1+2)/(11+3) = 3/14$

$p(book|a) = (1+1)/(11+2) = 2/13$

$p(<EOS>|book)= (1+1)/(11+2) = 2/13$

$$p(w_i \mid w_{i-1}) = \frac{1+c(w_{i-1}w_i)}{|V|+\sum_{w_i} c(w_{i-1}w_i)}$$

$$p(Cher\ read\ a\ book) = \frac{1}{14}\times\frac{1}{12}\times\frac{3}{14}\times\frac{2}{13}\times\frac{2}{13} \approx 0.00003$$

为避免概率值过小，可以改求log$p(Cher\ read\ a\ book)$

5.1.3 数据平滑方法

同理，其它 bi-grams 的概率变为：

$p(John|<BOS>) = 2/14,$

$p(read|John) = 2/12,$

$p(a|read) = 3/14,$

$p(book|a) = 2/13,$

$p(<EOS>|book) = 2/13$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

于是，

$$p(John\ read\ a\ book)$$
$$= p(John|<BOS>) \times p(read|John) \times p(a|read) \times p(book|a) \times p(<EOS>|book)$$
$$= \frac{2}{14} \times \frac{2}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13}$$
$$\approx 0.0001$$

平滑前为0.06。见P19。

5.1.3 数据平滑方法

(2) 减值法/折扣法

基本思想：修改训练样本中事件(N-gram)的实际计数，减小已出现事件的概率之和（<1），剩余的概率量分配给未见事件。

① Good-Turing 估计法

② 后备/后退法 (back-off)

③ 绝对减值法 (absolute discounting)

④ 线性减值法 (linear discounting)

5.1.3 数据平滑方法

①Good-Turing 估计法

对于任何一个发生r次的事件（N-gram），都假设它发生r*次：

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

n_r表示出现r次的N元对的个数，N-gram中出现次数为r的N元对 w_{i-n+1}^{i-1} 的出现概率为：

$$P_{GT}(w_{i-n+1}^{i}) = \frac{r^*}{\sum_{r=0}^{\infty} r^*}$$

Good-Turing 估计公式中缺乏利用低元模型对高元模型进行插值的思想，它通常不单独使用，而作为其他平滑算法中的一个计算工具。

5.1.3 数据平滑方法

举例说明：假设有如下英语文本，估计 2-gram概率：

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

<BOS>She read a book by Cher<EOS>

……

从训练集中统计出不同 2-grams 出现的次数分别：

<BOS>John	*7*
<BOS>Mary	10
……	
read Moby	5
read a	5
……	

5.1.3 数据平滑方法

假设要估计以 read 开始的 2-grams 概率，列出以read开始的所有 2-grams，并转化为频率信息：

R(次)	n_r(个数)	r^*
0	123544	
1	2053	0.446
2	458	1.25
3	191	2.24
4	107	3.22
5	69	4.17
6	48	5.25
7	36	保持原来的计数

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

$$=2\times 458/2053$$

$$=0.446$$

$$r^*=3\times 191/458$$

$$=1.25$$

……

R很大时n_{r+1}可能为 0

意味着：所有以read开始的2-gram，只要其原频率为2，均减为1.25…，以此类推

5.1.3 数据平滑方法

得到 r^* 后，就可以应用公式(5-7) 计算概率：

$$p_r = \frac{r^*}{N} \qquad \ldots (5\text{-}7)$$

其中，N 是以 read 开始的 bigrams 的总数(样本空间)，即 read 出现的次数。

那么，以 read开始、没有出现过的bigrams的概率总和为：

$$p_0 = \frac{n_1}{N}$$

利用N总数不变可推导出

意味着：剩余n_1/N的概率量分配给了所有未见事件(r=0的事件)

5.1.3 数据平滑方法

以 read开始、没有出现过的 bigrams的个数等于：

$$n_0 = |V_T| - \sum_{r>0} n_r$$ 其中，$|V_T|$ 为语料的词汇量。

于是，未见的那些以read 为开始的bigrams的概率平均为：$\frac{p_0}{n_0}$。

注意：$\sum_{r=0}^{7} p_r \neq 1$

因此，需要归一化处理：

$$\hat{p}_r = \frac{p_r}{\sum_r p_r}$$

Count	Count of counts	Adjusted count
r	N_r	r^*
0	7,514,941,065	0.00015
1	1,132,844	0.46539
2	263,611	1.40679
3	123,615	2.38767
4	73,788	3.33753
5	49,254	4.36967
6	35,869	5.32928
8	21,693	7.43798
10	14,880	9.31304
20	4,546	19.54487

5.1.3 数据平滑方法

② 后备/后退(Back-off)方法

S. M. Katz 于 1987 年提出，所以又称 Katz 后退法。

基本思想：当某一事件在样本中出现的频率大于阈值K (通常取 K 为0 或1)时，运用最大似然估计的减值法来估计其概率，否则，使用低阶的，即(n-1)gram 的概率替代 n-gram 概率，但这种替代需受归一化因子α的作用。

5.1.3 数据平滑方法

以2元文法模型为例, 说明Katz平滑方法：

对于一个出现次数为 $r = c(w_{i-1}^{i})$ 的 2元文法 w_{i-1}^{i}，使用如下公式计算修正的概率：

$$p_{\text{katz}}(w_i \mid w_{i-1}) = \begin{cases} d_r \dfrac{C(w_{i-1}w_i)}{C(w_{i-1})} & \text{if } C(w_{i-1}w_i) = r > 0 \\ \alpha(w_{i-1})\, p_{\text{ML}}(w_i) & \text{if } C(w_{i-1}w_i) = 0 \end{cases}$$

低阶概率替代

其中，$p_{\text{ML}}(w_i)$表示 w_i 的最大似然估计概率。这个公式的意思是，所有非零计数 r 的 2元文法都根据折扣率d_r $(0<d_r<1)$被减值，折扣率 $d_r = r^*/r$，r^* 由 Good-Turing 法预测。

5.1.3 数据平滑方法

那么，如何确定$\alpha(w_{i-1})$呢？

$$\sum_{w_i} p_{\text{katz}}(w_i|w_{i-1}) = 1$$

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

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

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

5.1.3 数据平滑方法

③ 绝对减值法 (Absolute discounting)

基本思想：从每个计数 r 中减去同样的量，剩余的概率量由未见事件均分。

设 R 为所有可能事件的数目(对 n-gram ， 如词汇集的大小为 L, 则 $R=L^n$)。

5.1.3 数据平滑方法

那么，样本出现了 r 次的事件的概率可以由如下公式估计：

$$p_r = \begin{cases} \dfrac{r-b}{N} & \text{当 } r > 0 \\ \dfrac{b(R-n_0)}{Nn_0} & \text{当 } r = 0 \end{cases} \quad \text{由}n_0\text{个事件均分} \qquad \ldots (5\text{-}10)$$

其中，n_0 为样本中未出现的事件的数目。b 为减去的常量，$b \leq 1$。$b(R - n_0)/N$ 是由于减值而产生的概率量。N 为样本中出现了r 次的事件总次数：$n_r \times r$。

5.1.3 数据平滑方法

b 为自由参数，可以通过留存数据(heldout data)方法求得 b 的上限为：

$$b \le \frac{n_1}{n_1 + 2n_2} < 1 \qquad \ldots (5\text{-}11)$$

留存数据法：训练数据分为两部分，一部分用于计算初始概率，另一部分留出来用于计算自由参数，改善初始计算出来的概率。

5.1.3 数据平滑方法

④ 线性减值法 (Linear discounting)

基本思想：从每个计数 r 中减去与该计数成正比的量(减值函数为线性的)，剩余概率量α 被 n_0个未见事件均分。

$$p_r = \begin{cases} \dfrac{(1-\alpha)r}{N} & \text{当 } r > 0 \\ \dfrac{\alpha}{n_0} & \text{当 } r = 0 \end{cases} \quad \ldots (5\text{-}12)$$

自由参数 α 的优化值为：$\dfrac{n_1}{N}$。参见 Good-Turing 法。

在很多实验中，绝对减值法产生的 n-gram 优于线性减值法。

5.1.3 数据平滑方法

- 4种减值法的比较
  - Good-Turing 法：对非0事件按公式削减出现的次数，节留出来的概率均分给0概率事件。
  - Katz 后退法：对非0事件按Good-Turing法计算减值，节留出来的概率按低阶分布分给0概率事件。
  - 绝对减值法：对非0事件无条件削减某一固定的出现次数值，节留出来的概率均分给0概率事件。
  - 线性减值法：对非0事件根据出现次数按比例削减次数值，节留出来的概率均分给0概率事件。

5.1.3 数据平滑方法

(3) 删除插值法 (Deleted interpolation)

也称为Jelinek-Mercer平滑。

基本思想：利用低阶N-gram模型对高阶N-gram 模型进行线性插值。如对于3-gram的概率值，可以将其与2-gram和unigram的概率值进行插值计算。插值公式：

$$p(w_3 \mid w_1 w_2) = \lambda_3 p'(w_3 \mid w_1 w_2) + \lambda_2 p'(w_3 \mid w_2) + \lambda_1 p'(w_3) \quad \ldots (5\text{-}13)$$

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

极大似然估计

5.1.3 数据平滑方法

(3) 删除插值法 (Deleted interpolation)

插值法的递归定义如下

$$P_{interp}(w_i|w_{i-(n-1)}\cdots w_{i-1}) = \lambda P_{ML}(w_i|w_{i-(n-1)}\cdots w_{i-1}) + (1-\lambda)P_{interp}(w_i|w_{i-(n-2)}\cdots w_{i-1})$$

为了结束递归，可以令Unigram模型为极大似然估计模型，或者为均匀分布模型

$$p_{o阶}(w_i) = \frac{1}{|V|}$$

5.1.3 数据平滑方法

- $\lambda_1, \lambda_2, \lambda_3$ 的确定：

将训练语料分为两部分：一部分用于计算初始概率：$p'(w_3 \mid w_1 w_2)$，$p'(w_3 \mid w_2)$ 和 $p'(w_3)$；另一部分作为留存数据用于估计λ_1，λ_2，λ_3，其目标是在留存数据上使语言模型的困惑度最小。

5.1.3 数据平滑方法

- 各种平滑方法的详细介绍和比较请参阅：

  Stanley F. Chen and Joshua T. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling, Technical Report TR-10-98, Computer Science Group, Harvard University
  http://www-2.cs.cmu.edu/~sfc/html/publications.html

- SRI 语言模型：

  http://www.speech.sri.com/projects/srilm/

- CMU-Cambridge 语言模型：

  http://mi.eng.cam.ac.uk/~prc14/toolkit.html

5.1 传统语言模型

5.1.1 n元文法

5.1.2 参数估计

5.1.3 数据平滑方法

5.1.4 语言模型自适应

5.1.5 n元文法模型的应用

5.4 语言模型自适应

问题：

①在训练语言模型时所采用的语料往往来自多种不同的领域，这些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，而语言模型恰恰对于训练样本的类型、主题和风格等都十分敏感；

②n 元文法模型独立性假设的前提是，文本中当前词出现的概率只与它前面相邻的 n-1 个词相关，但这种假设在很多情况下是明显不成立的。

目的：提高语言模型对语料的领域、主题、类型等因素的适应性

5.4 语言模型自适应

自适应方法：

(1)基于缓存的语言模型 (cache-based LM)

文本中刚刚出现过的一些词在后边的句子中再次出现的可能性往往较大

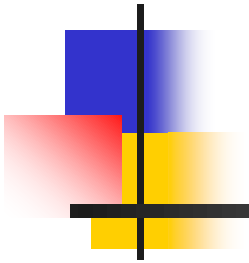
(2)基于混合方法的语言模型

为适应n种语料对语言模型性能的影响，将语言模型划分成n个特定的子模型，再通过线性插值集成。

(3)基于最大熵的语言模型

每种语料构建一个语言模型，并提供一组约束条件，然后在所有满足约束的模型中，选择熵最大的模型。

课后阅读



5.1 传统语言模型

5.1.1 n元文法

5.1.2 参数估计

5.1.3 数据平滑方法

5.1.4 语言模型自适应

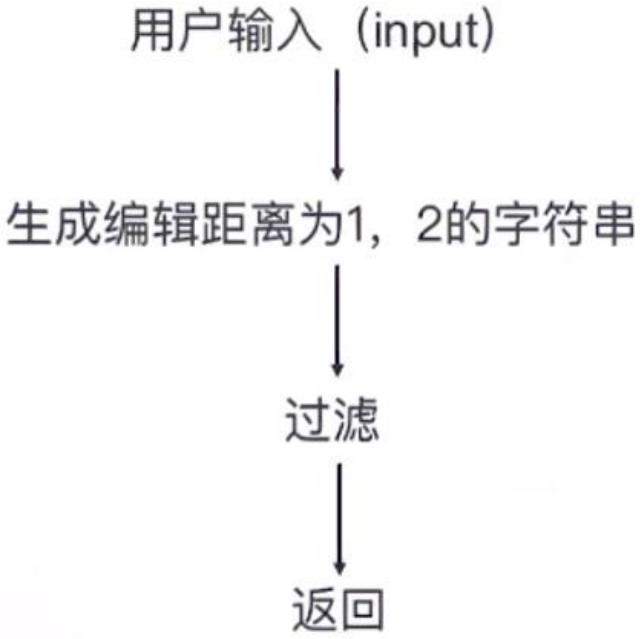
5.1.5 n元文法模型的应用

5.1.5 n元文法模型的应用

◆以拼写纠错为例

句子：My favourite foot is apple。 应纠错为food

基本思路：



5.1.5 n元文法模型的应用

◆以拼写纠错为例

如何生成候选词

编辑距离是指：给定字符串str1和str2, 将str1转换成str2所需要的代价。

1. 插入新字符 2. 替换字符 3. 删除一个字符。 每一个操作的代价为1.

5.1.5 n元文法模型的应用

◆以拼写纠错为例

如何过滤

假设一句话w中含有错误单词w_i，我们要将其改为正确的语句s(将w_i替换为s_i)，可将其建模为下列问题：

$$s^* = \underset{s}{\operatorname{argmax}} P(s|w) = \underset{s}{\operatorname{argmax}} \frac{P(w|s)P(s)}{p(w)} = \underset{s}{\operatorname{argmax}} P(w|s)P(s)$$

$$= \underset{s}{\operatorname{argmax}} P(w_i|s_i)P(s)$$

$p(w_i|s_i)$衡量备选词si和wi的相关性，其值可以由编辑距离(或转移概率统计)来决定。一般编辑距离越大，其概率越小。

5.1.5 n元文法模型的应用

◆以拼写纠错为例

如何过滤

$$\text{argmax}P(w_i|s_i)P(s)$$

p(s)衡量将备选词s_i放到语言模型中的概率。以food为例，将food放入原句，得到 My favourite food is apple，用语言模型去衡量这个句子是否通顺，即：

p(s)~p(My)p(favourite｜My)p(food｜favourite)p(is｜food)p(apple｜is)

5.1.5 n元文法模型的应用

◆以汉语分词和命名实体识别为例

句子：这篇文章写得太平淡了。

这/ 篇/ 文章/ 写/ 得/ 太/ 平淡/ 了/ 。

这/ 篇/ 文章/ 写/ 得/ 太平/ 淡/ 了/ 。

5.1.5 n元文法模型的应用

采用基于语言模型的分词方法

- 方法描述：

设对于待切分的句子$S = z_1 z_2 \ldots z_m$，$W = w_1 w_2 \ldots w_N$ $(1 \le N \le m)$是一种可能的切分。那么，

$$\hat{W} = \arg\max_{W} p(W \mid S)$$

$$= \arg\max_{W} p(W) \times p(S \mid W) \qquad \ldots (5\text{-}22)$$

5.1.5 n元文法模型的应用

具体实现时，可以把汉语词汇分成如下几类：

(1) 分词词典中规定的词；

(2) 可以由词法规则派生出来的词或短语，如：干干净净、非党员、副部长、全面性、检查员、看不出、克服了、走出来 、洗个澡 …

(3) 与数字相关的实体，如：日期、时间、货币、百分数、温度、长度、面积、重量、电话号码、邮件地址等；

(4) 专用名词，如 ：人名、地名、组织机构名。

占未登录词的95%！

5.1.5 n元文法模型的应用

进一步做如下约定，把一个可能的词序列 W 转换成词类序列 $C = c_1 c_2 \cdots c_N$ ，即：

- 专有名词：人名PN、地名LN、机构名ON分别作为一类；
- 实体名词中的日期dat、时间tim、百分数per、货币mon、型号typ等分别作为一类；
- 由词法规则派生出来的词和词表中的词，每个词单独作为一类。

5.1.5 n元文法模型的应用

例如：

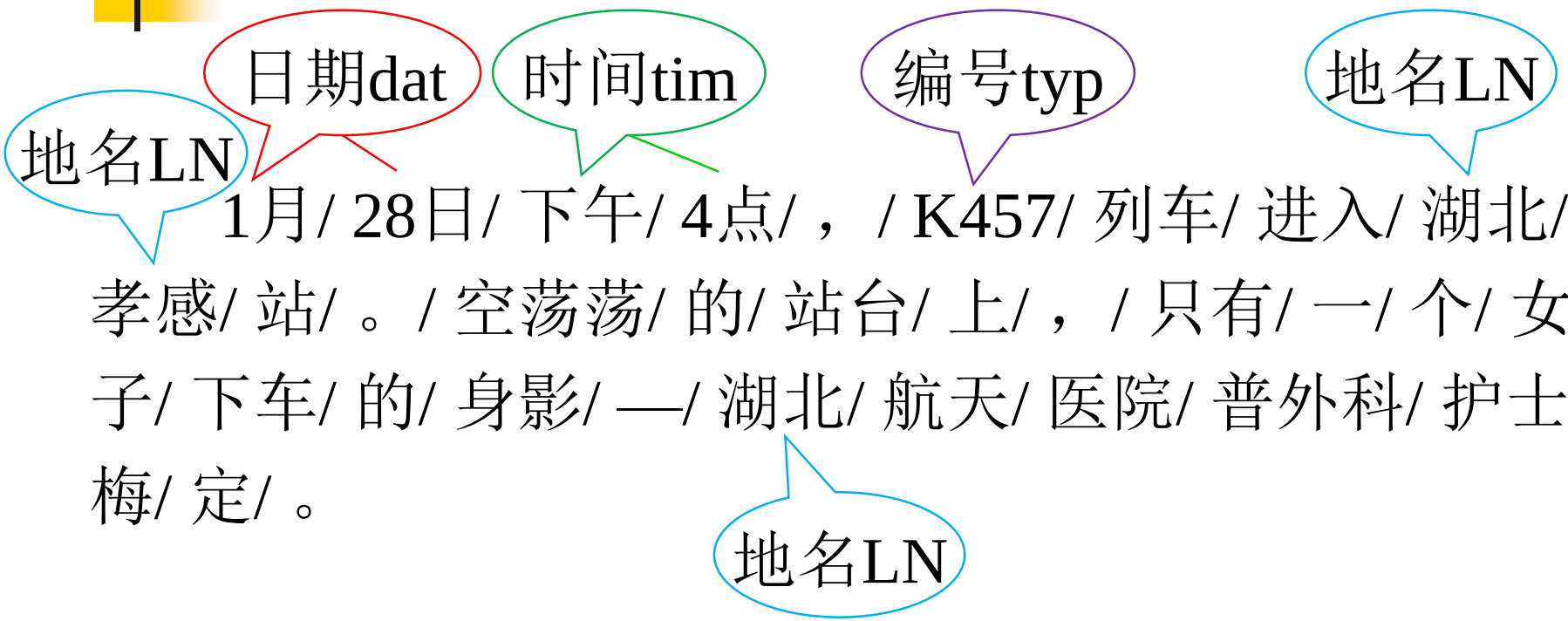
1月28日下午4点，K457列车进入湖北孝感站。空荡荡的站台上，只有一个女子下车的身影—湖北航天医院普外科护士梅定。

5.1.5 n元文法模型的应用

分词结果：

1月/ 28日/ 下午/ 4点/ ，/ K457/ 列车/ 进入/ 湖北/ 孝感/ 站/ 。/ 空荡荡/ 的/ 站台/ 上/ ，/ 只有/ 一/ 个/ 女子/ 下车/ 的/ 身影/ —/ 湖北/ 航天/ 医院/ 普外科/ 护士/ 梅/ 定/ 。

5.1.5 n元文法模型的应用



5.1.5 n元文法模型的应用

用词类替换原词，样本变成词类和词混合的序列：

dat1/ dat2/ tim1/ tim2/ ，/ typ/ 列车/ 进入/ LN1/ LN2/ 站/ 。/ 空荡荡/ 的/ 站台/ 上/ ，/ 只有/ 一/ 个/ 女子/ 下车/ 的/ 身影/ —/ LN1/ 航天/ 医院/ 普外科/ 护士/ 梅/ 定/ 。

由词法规则派生出来的词和词表中的词也可以替换。

5.1.5 n元文法模型的应用

同时进行命名实体识别的结果：

[1月/28日] [下午/4点]，/ K457/ 列车/ 进入/ [湖北/孝感] 站/ 。/ 空荡荡/ 的/ 站台/ 上/ ，/ 只有/ 一/ 个/ 女子/ 下车/ 的/ 身影/ —/ [湖北/航天/医院] 普外科/ 护士/ [梅/定] 。

词类替换后的结果：

dat/ tim/ ，/ typ/ 列车/ 进入/ LN/ 站/ 。/ 空荡荡/ 的/ 站台/ 上/ ，/ 只有/ 一/ 个/ 女子/ 下车/ 的/ 身影/ —/ ON/ 普外科/ 护士/ PN/ 。

5.1.5 n元文法模型的应用

那么，根据(5-22)式：

$$\hat{W} = \arg\max_{W} p(W \mid S) ; \quad \arg\max_{W} p(W) \times p(S \mid W) \quad \ldots (5\text{-}22)$$

$$; \quad \arg\max_{C} p(C) \times p(S \mid C)$$

语言模型　　生成模型

$p(C)$可采用trigram计算：

$$p(C) = p(c_1) \times p(c_2 \mid c_1) \prod_{i=3}^{N} p(c_i \mid c_{i-2} c_{i-1}) \quad \ldots (5\text{-}23)$$

$$p(c_i \mid c_{i-2} c_{i-1}) = \frac{count(c_{i-2} c_{i-1} c_i)}{count(c_{i-2} c_{i-1})} \quad \ldots (5\text{-}24)$$

5.1.5 n元文法模型的应用

生成模型在满足独立性假设的条件下，可近似为：

$$p(S \mid C) \approx \prod_{i=1}^{N} p(w_i \mid c_i) \qquad \ldots (5\text{-}25)$$

该公式的含意是：任意一个词类 c_i 生成汉字串 w_i 的概率只与自身有关，而与其上下文无关。

对于不同类别的词, 分别计算其概率。除了人名、地名和组织机构名称以外，如果某个词属于某一类，如“学生”属于词表词(LW)，令：$p(w_i=\text{学生}|c_i=\text{LW})=1$。

5.1.5 n元文法模型的应用

词 类 (C)	生成模型 $p(w_i\|C)$	语言知识来源
词表词 (LW)	若w_i是词表词，$p(w_i\|\text{LW})=1$，否则，判断是否其它类别；	分词词表
词法派生词 (MW)	若w_i是派生词，$p(w_i\|\text{MW})=1$，否则，判断是否其它类别；	派生词词表
人名 (PN)	基于字的2元模型	姓氏表，中文人名模板
地名 (LN)	基于字的2元模型	地名表、地名关键词表、地名简称表
机构名 (ON)	基于词类的2元模型	机关名关键词表，机构名简称表
其他实体名 (FT)	若w_i可用实体名词规则集G识别，$p(S\|G)=1$，否则，判断是否其它类别。	实体名词规则集

5.1.5 n元文法模型的应用

模型的训练由以下三步组成：

(1) 在词表和派生词表的基础上，用一个基本的分词工具切分训练语料；专有名词通过一个专门模块标注，实体名词通过相应的规则和有限状态自动机标注，由此产生一个带词类别标记的初始语料；

(2) 用带词类别标记的初始语料，采用最大似然估计方法估计语言模型的概率参数，公式(5-24)；

(3) 用得到的模型（公式(5-22)、(5-23)、(5-25)）对训练语料重新切分和标注，得到新的训练语料；

(4) 重复(2)(3)步，直到系统的性能不再有明显的变化为止。

5.1.5 n元文法模型的应用

- 实验语料：

(1)词表词：98,668条、派生词：59,285条；

(2)训练语料：88MB 新闻文本；

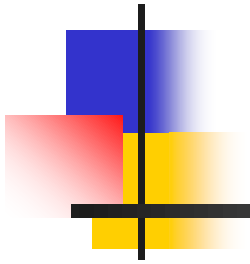
(3)测试集：247,039个词次，分别来自描写文、叙述文、说明文、口语等。

- 测试指标：

$$\text{正确率} = \frac{\text{切分正确的词数}}{\text{系统输出的总词数}} \times 100\% = 96.3\%$$

本部分小结

- 传统语言模型的基本概念
  n元文法, 马尔可夫链
- 参数估计
- 数据平滑方法：
  - 加1法
  - 减值法：1) Good-Turing; 2) Back-off (Katz);
    3) 绝对减值； 4) 线性减值
  - 删除插值法
- n元文法模型的自适应方法
- n元模型应用举例



Thanks

謝謝！