

第五章 语言模型

目录

- 5.1 基本概念
- 5.2 参数估计
- 5.3 数据平滑
- 5.4 语言模型的自适应
- 5.5 语言模型应用举例
- 5.6 神经网络语言模型

5.1 基本概念

5.1 基本概念

- 大规模语料库的出现为自然语言统计处理方法的实现提供了可能，同时统计方法的成功使用推动了语料库语言学的发展。
- 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推测

5.1 基本概念

◆ 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 基本概念

◆ 如何计算一段文字(句子)的概率?

语句 $s = w_1 w_2 \cdots w_n$ 的先验概率:

$$\begin{aligned} p(s) &= p(w_1) \times p(w_2|w_1) \times p(w_3|w_2 w_1) \times \cdots \\ &\quad \times p(w_m|w_1 \cdots w_{m-1}) \\ &= \prod_{i=1}^m p(w_i|w_1 \cdots w_{i-1}) \end{aligned} \quad (5-1)$$

当 $i = 1$ 时, $p(w_1|w_0) = p(w_1)$ 。

语言模型

5.1 基本概念

说明:

- (1) w_i 可以是字、词、短语或词类等等, 称为统计基元。
通常以“词”代之。
- (2) w_i 的概率由 w_1, \dots, w_{i-1} 决定, 由特定的一组 w_1, \dots, w_{i-1} 构成的一个序列, 称为 w_i 的历史 (history)。

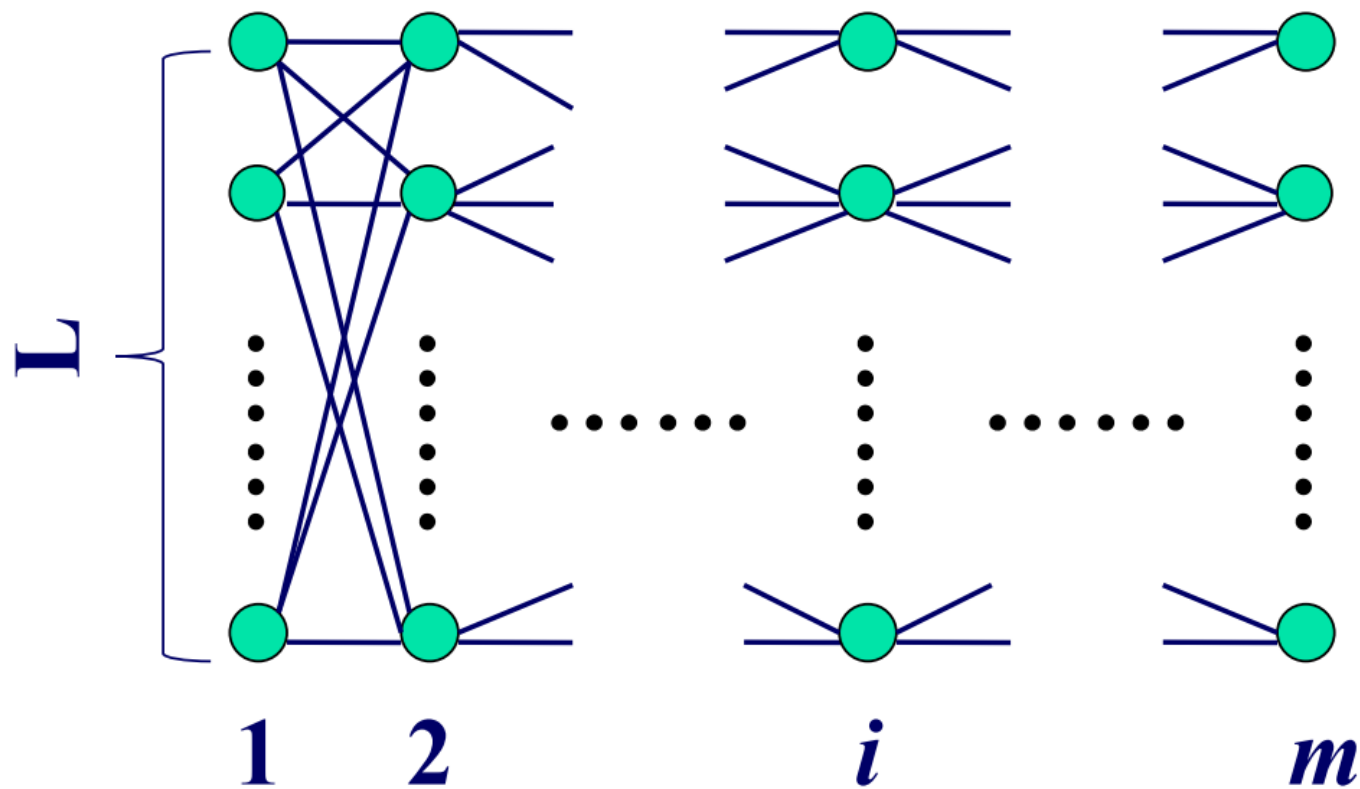
5.1 基本概念

问题：

随着历史基元数量的增加，不同的“历史”（路径）按指数级增长。对于第 $i (i > 1)$ 个统计基元，历史基元的个数为 $i - 1$ ，如果共有 L 个不同的基元，如词汇表，理论上每一个单词都有可能出现在 1 到 $i - 1$ 的每一个位置上，那么， i 基元就有 L^{i-1} 种不同的历史情况。我们必须考虑在所有的 L^{i-1} 种不同历史情况下产生第 i 个基元的概率。那么，模型中有 L^m 个自由参数

$$p(w_m | w_1 \cdots w_{m-1})$$

5.1 基本概念



如果 $L = 5000$, $m = 3$, 自由参数的数据为1250亿!

5.1 基本概念

◆ 问题解决方法

设法减少历史基元的个数，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：

$$p(w_i | w_1 \cdots w_{i-1}) = p(w_i | S(w_1 \cdots w_{i-1})) \quad (5-2)$$

5.1 基本概念

◆ 如何划分等价类

将两个历史映射到同一个等价类，当且仅当这两个历史中的最近 $n - 1$ 个基元相同，即：

$$\begin{array}{ccccccc} H_1: & w_1 & w_2 & \dots & \dots & w_{i-n+1} & w_{i-n+2} & \dots & w_{i-1} & w_i & \dots & \dots \\ & & & & & \underbrace{\hspace{10em}}_{n-1} & & & & \uparrow & & \\ & & & & & & & & & \downarrow & & \\ H_2: & v_1 & v_2 & \dots & \dots & v_{k-n+1} & v_{k-n+2} & \dots & v_{k-1} & v_k & \dots & \dots \end{array}$$

如果 $S(w_1, w_2, \dots, w_i) = S(v_1, v_2, \dots, v_k)$

iff $H_1: (w_{i-n+1}, \dots, w_i) = H_2: (v_{k-n+1}, \dots, v_k)$

(5-3)

5.1 基本概念

◆ n 元语法(n -gram)

这种情况下的语言模型称为 **n 元文法(n -gram)**。

通常地,

- 当 $n = 1$ 时, 即出现在第 i 位上的基元 w_i 独立于历史。一元文法也写为 uni-gram 或 monogram;
- 当 $n = 2$ 时, 2-gram(bi-gram)称为1阶马尔可夫链;
- 当 $n = 3$ 时, 3-gram(tri-gram)称为2阶马尔可夫链, 依次类推。

5.1 基本概念

◆ n 元语法(n-gram)

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

$$p(s) = \prod_{i=1}^{m+1} p(w_i | w_{i-n+1}^{i-1}) \quad (5-4)$$

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>**。

5.1 基本概念

举例：

给定句子： John read a book

增加标记： <BOS> John read a book <EOS>

Unigram: <BOS> , John, read, a, book, <EOS>

Bigram: (<BOS> John), (John read), (read a),
(a book), (book <EOS>)

Trigram: (<BOS> John read), (John read a),
(read a book), (a book <EOS>)

5.1 基本概念

举例：

<BOS> John read a book <EOS>

基于2元文法的概率为：

$$\begin{aligned} p(\text{John read a book}) &= p(\text{John}|\text{<BOS>}) \times \\ &\quad p(\text{read}|\text{John}) \times p(\text{a}|\text{read}) \times \\ &\quad p(\text{book}|\text{a}) \times p(\text{<EOS>}|\text{book}) \end{aligned}$$

5.1 基本概念

◆ 应用-1：音字转换问题

给定拼音串： **ta shi yan jiu sheng wu de**

可能的汉字串： 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

... ..

5.1 基本概念

◆应用-1：音字转换问题

$$\begin{aligned}\hat{CString} &= \arg \max_{CString} p(CString | Pinyin) \\ &= \arg \max_{CString} \frac{p(Pinyin | CString) \times p(CString)}{p(Pinyin)} \\ &= \arg \max_{CString} p(Pinyin | CString) \times p(CString) \\ &= \arg \max_{CString} p(CString)\end{aligned}$$

5.1 基本概念

◆应用-1：音字转换问题

$CString = \{\text{踏实研究生物的, 他实验救生物的, 他是研究生物的, 他使烟酒生雾的, ...}\}$

如果使用 2-gram:

$$p(CString_1) = p(\text{踏实} | \langle \text{BOS} \rangle) \times p(\text{研究} | \text{踏实}) \times \\ p(\text{生物} | \text{研究}) \times p(\text{的} | \text{生物}) \times p(\langle \text{EOS} \rangle | \text{的})$$

$$p(CString_2) = p(\text{他} | \langle \text{BOS} \rangle) \times p(\text{实验} | \text{他}) \times p(\text{救} | \text{实验}) \times \\ p(\text{生物} | \text{救}) \times p(\text{的} | \text{生物}) \times p(\langle \text{EOS} \rangle | \text{的})$$

.....

5.1 基本概念

◆ 应用-1：音字转换问题

如果汉字的总数为： N

- 一元语法：
 - 1) 样本空间为 N
 - 2) 只选择使用频率最高的汉字
- 2元语法：
 - 1) 样本空间为 N^2
 - 2) 效果比一元语法明显提高
- 估计对汉字而言四元语法效果会好一些
- 智能狂拼、微软拼音输入法基于 **n-gram**.

5.1 基本概念

◆应用-2：汉语分词问题

给定汉字串：他是研究生物的。

可能的汉字串：

1) 他|是|研究生|物|的

2) 他|是|研究|生物|的

5.1 基本概念

◆应用-2：汉语分词问题

$$\begin{aligned}\widehat{Seg} &= \arg \max_{Seg} p(Seg | Text) \\ &= \arg \max_{Seg} \frac{p(Text | Seg) \times p(Seg)}{p(Text)} \\ &= \arg \max_{Seg} p(Text | Seg) \times p(Seg) \\ &= \arg \max_{Seg} p(Seg)\end{aligned}$$

5.1 基本概念

◆ 应用-2: 汉语分词问题

如果采用 2-gram:

$$p(\text{Seg1}) = p(\text{他} | \langle \text{BOS} \rangle) \times p(\text{是} | \text{他}) \times p(\text{研究生} | \text{是}) \times \\ p(\text{物} | \text{研究生}) \times p(\text{的} | \text{物}) \times p(\text{的} | \langle \text{EOS} \rangle)$$

$$p(\text{Seg2}) = p(\text{他} | \langle \text{BOS} \rangle) \times p(\text{是} | \text{他}) \times p(\text{研究} | \text{是}) \times \\ p(\text{生物} | \text{研究}) \times p(\text{的} | \text{生物}) \times \\ p(\text{的} | \langle \text{EOS} \rangle)$$

问题: 如何获得 n 元语法模型?

5.2 参数估计

5.2 参数估计

◆ 两个重要概念:

- **训练语料(training data)**: 用于建立模型, 确定模型参数的已知语料。
- **最大似然估计(maximum likelihood Evaluation, MLE)**: 用相对频率计算概率的方法。

5.2 参数估计

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:

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

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

5.2 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？

5.2 参数估计

$$p(\text{John} | \langle \text{BOS} \rangle) = \frac{c(\langle \text{BOS} \rangle \text{John})}{\sum_w c(\langle \text{BOS} \rangle w)} = \frac{1}{3} \quad p(a | \text{read}) = \frac{c(\text{read } a)}{\sum_w c(\text{read } w)} = \frac{2}{3}$$

$$p(\text{read} | \text{John}) = \frac{c(\text{John } \text{read})}{\sum_w c(\text{John } w)} = \frac{1}{1} \quad p(\text{book} | a) = \frac{c(a \text{ book})}{\sum_w c(a w)} = \frac{1}{2}$$

$$p(\langle \text{EOS} \rangle | \text{book}) = \frac{c(\text{book } \langle \text{EOS} \rangle)}{\sum_w c(\text{book } w)} = \frac{1}{2}$$

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

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

5.2 参数估计

$$p(\text{Cher read a book}) = ?$$

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

$$p(\text{Cher} | \langle \text{BOS} \rangle) = \frac{c(\langle \text{BOS} \rangle \text{ Cher})}{\sum_w c(\langle \text{BOS} \rangle w)} = \frac{0}{3}$$

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



于是, $p(\text{Cher read a book}) = 0$

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

5.2 参数估计

问题:

数据匮乏(稀疏) (Sparse Data) 引起零概率问题,
如何解决?



数据平滑(data smoothing)

5.3 数据平滑

5.3 数据平滑

◆ 数据平滑的基本思想：

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，“劫富济贫”，消除零概率，改进模型的整体正确率。

◆ 基本目标：

测试样本的语言模型困惑度越小越好。

◆ 基本约束：

$$\sum_{w_i} p(w_i | w_1, w_2, \dots, w_{i-1}) = 1$$

5.3 数据平滑

◆ 回顾—困惑度的定义：

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:

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

假定测试语料 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:

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

5.3 数据平滑

◆ 回顾—困惑度的定义：

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 是测试文本 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 对英语文本的困惑度范围一般为 50~1000，
对应于交叉熵范围为 6 ~ 10 *bits/word*。

5.3 数据平滑

◆ 加 1 法(Additive smoothing):

基本思想: 每一种情况出现的次数加1。

例如, 对于uni-gram, 设 w_1, w_2, w_3 三个词, 概率分别为: $1/3, 0, 2/3$, 加 1 后情况?

$2/6, 1/6, 3/6$

5.3 数据平滑

◆ 加 1 法(Additive smoothing):

对于2-gram 有:

$$\begin{aligned} p(w_i|w_{i-1}) &= \frac{1 + c(w_{i-1}w_i)}{\sum_{w_i} [1 + c(w_{i-1}w_i)]} \\ &= \frac{1 + c(w_{i-1}w_i)}{|V| + \sum_{w_i} c(w_{i-1}w_i)} \end{aligned}$$

其中, V 为被考虑语料的词汇量 (全部可能的基元数) 。

5.3 数据平滑

◆ 加 1 法(Additive smoothing):

在前面 3 个句子的例子中,

$p(\textit{Cher read a book})$

$$= p(\textit{Cher} | \langle \textit{BOS} \rangle)$$

$$\times p(\textit{read} | \textit{Cher}) \times p(\textit{a} | \textit{read}) \times p(\textit{book} | \textit{a})$$

$$\times p(\langle \textit{EOS} \rangle | \textit{book})$$

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

原来:

$$p(\textit{Cher} | \langle \textit{BOS} \rangle) = 0/3$$

$$p(\textit{read} | \textit{Cher}) = 0/1$$

$$p(\textit{a} | \textit{read}) = 2/3$$

$$p(\textit{book} | \textit{a}) = 1/2$$

$$p(\langle \textit{EOS} \rangle | \textit{book}) = 1/2$$

5.3 数据平滑

◆ 加 1 法(Additive smoothing):

词汇量: $|V| = 11$

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

平滑以后:

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

$$p(\text{read} | \text{Cher}) = (0 + 1) / (11 + 1) = 1/12$$

$$p(a | \text{read}) = (1 + 2) / (11 + 3) = 3/14$$

$$p(\text{book} | a) = (1 + 1) / (11 + 2) = 2/13$$

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

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

5.3 数据平滑

◆ 加 1 法(Additive smoothing):

同理, 对于句子 John read a book 数据平滑后:

$$p(\text{John} | \langle \text{BOS} \rangle) = 2/14, \quad p(\text{read} | \text{John}) = 2/12,$$

$$p(a | \text{read}) = 3/14, \quad p(\text{book} | a) = 2/13,$$

$$p(\langle \text{EOS} \rangle | \text{book}) = 2/13$$

$$\begin{aligned} \text{于是 } p(\text{John read a book}) &= p(\text{John} | \langle \text{BOS} \rangle) \times p(\text{read} | \text{John}) \\ &\quad \times p(a | \text{read}) \times p(\text{book} | a) \times p(\langle \text{EOS} \rangle | \text{book}) \\ &= \frac{2}{14} \times \frac{2}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx \mathbf{0.0001} \end{aligned}$$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

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

5.3 数据平滑

◆ 减值法/折扣法 (Discounting):

基本思想: 修改训练样本中事件的实际计数, 使样本中(实际出现的)不同事件的概率之和小于1, 剩余的概率量分配给未见事件。

5.3 数据平滑

◆ 减值法/折扣法 (Discounting):

(1) Good-Turing 估计

I. J. Good于1953年引用Turing的方法来估计概率分布。

基本思想：将统计参数按出现次数聚类，然后用出现次数加一的类来估计当前类。

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

假设 N 是原来训练样本数据的大小, n_r 是在样本中正好出现 r 次的事件的数目(此处事件为 n -gram), 即出现 1 次的 n -gram 有 n_1 个, 出现 2 次的 n -gram 有 n_2 个,, 那么

$$N = \sum_{r=1}^{\infty} n_r r \quad (5-6)$$

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

由于, $N = \sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1)n_{r+1} = \sum_{r=1}^{\infty} n_r r$

所以Good-Turing平滑计数:

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

那么, Good-Turing 估计在样本中出现 r 次的事件的概率为:

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

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

实际应用中，一般直接用 n_{r+1} 代替 $E(n_{r+1})$, n_r 代替 $E(n_r)$ 。则原训练样本中所有事件的概率之和为：

$$\sum_{r>0} n_r \times p_r = 1 - \frac{n_1}{N} < 1 \quad (5-8)$$

因此，有 $\frac{n_1}{N}$ 的剩余的概率量可以均分给所有的未见事件 ($r = 0$ 的事件)。

Good-Turing 估计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。

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

举例说明：假设有如下英语文本，估计bigram概率：

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

从文本中统计出不同 bigram 出现的次数：

<BOS> John 15

<BOS> Mary 10

.....

read Moby 5

.....

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

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

r	n_r	r^*
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

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

$$n_{r+1} = 0$$

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

得到 r^* 后，就可以应用公式(5-7) 计算概率：

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

其中， N 为以read开始的bigram的总数(样本空间)，即read出现的次数。则以read开始，没出现过的bigram的概率总和为： $p_0 = \frac{n_1}{N}$

以read作为开始，没有出现过的bigram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。

5.3 数据平滑

◆ 减值法/折扣法——Good-Turing 估计

那么，没有出现过的那些 以read为开始的bigram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：

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

r	n_r	r^*
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	—

5.3 数据平滑

◆ 减值法/折扣法 (Discounting):

(2) Back-off (后备/后退)方法

S. M. Katz 于1987年提出，所以又称 Katz 平滑方法。

基本思想：当某一事件在样本中出现的频率大于 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，而这种替代需受归一化因子 α 的作用。

5.3 数据平滑

◆ 减值法/折扣——Back-off (后备/后退)方法

Back-off 方法的另一种解释:

对于每个计数 $r > 0$ 的 n -gram 的出现次数减值, 把因减值而节省下来的剩余概率根据低阶的 $(n-1)$ gram 分配给未见事件。

5.3 数据平滑

◆ 减值法/折扣——Back-off (后备/后退)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bigram w_{i-1}^i ，使用如下公式计算修正的概率：

$$p_{katz}(w_i | w_{i-1}^i) = \begin{cases} d_r \frac{c(w_{i-1}^i)}{c(w_{i-1})} & \text{if } r > 0 \\ \alpha(w_{i-1}) p_{ML}(w_i) & \text{if } r = 0 \end{cases}$$

bigram

unigram

$p_{ML}(w_i)$ 表示 w_i 的最大似然估计概率。

第一种解释

5.3 数据平滑

◆ 减值法/折扣——Back-off (后备/后退)方法

这个公式的意思是：所有具有非零计数 r 的bigram都根据折扣率 $d_r (0 < d_r < 1)$ 被减值了，折扣率 d_r 近似等于 r^*/r ，减值由Good-Turing估计方法预测。

从非零计数中减去的计数量，根据低一阶的分布，即一元文法模型被分配给了计数为零的bigram。则需选择 $\alpha(w_{i-1})$ 值，使分布中总的计数 $\sum_{w_i} c_{katz}(w_{i-1}^i)$ 保持不变，即

$$\sum_{w_i} c_{katz}(w_{i-1}^i) = \sum_{w_i} c(w_{i-1}^i)$$

5.3 数据平滑

◆ 减值法/折扣——Back-off (后备/后退)方法

那么，如何确定 $\alpha(w_{i-1})$ 的值？

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



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



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



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

5.3 数据平滑

◆ 减值法/折扣——Back-off (后备/后退)方法

$\alpha(w_{i-1})$ 适当的取值为:

$$d_r r / N$$

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

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

5.3 数据平滑

◆ 减值法/折扣——Back-off (后备/后退)方法

另一种解释: (可取 $K = 0$)

$c(\bullet)$ 是指“ \bullet ”出现的频次

最大似然估计方法
求 $n - gram$ 概率

$$p(w_i | w_{i-n+1}^{i-1}) = \begin{cases} \left(1 - \alpha(c(w_{i-n+1}^i))\right) \frac{c(w_{i-n+1}^i)}{c(w_{i-n+1}^{i-1})} & \text{if } c(w_{i-n+1}^i) > K \\ \alpha(c(w_{i-n+1}^{i-1})) p(w_i | w_{i-n+2}^{i-1}) & \text{if } c(w_{i-n+1}^i) \leq K \end{cases}$$

α 是归一化因子,
为频次 c 的函数

$(n - 1) gram$ 概率

(5-9)

5.3 数据平滑

◆ 减值法/折扣法 (Discounting):

(3) 绝对减值法 (Absolute discounting)

Hermann Ney 和 U. Essen 1993年提出。

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（当事件为 n -gram 时，
如果统计基元为词，且词汇集的大小为 L ，则 $R = L^n$ ）。

5.3 数据平滑

◆ 减值法/折扣法——绝对减值法

则样本出现了 r 次的事件的概率可以由如下公式估计：

$$p_r = \begin{cases} \frac{r-b}{N} & \text{当 } r > 0 \\ \frac{b(R-n_0)}{Nn_0} & \text{当 } r = 0 \end{cases} \quad (5-10)$$

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剩余概率量。

5.3 数据平滑

◆ 减值法/折扣法——绝对减值法

b 为自由参数，可以通过留存数据(heldout data)法求得，利用留一法(leave one out)可以求得 b 的上限为：

$$b \leq \frac{n_1}{n_1 + 2n_2} < 1 \quad (5-10)$$

实际运用中，常用上限代替优化的 b 。

5.3 数据平滑

◆ 减值法/折扣法 (Discounting):

(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。

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

自由参数 α 的优化值为: $\frac{n_1}{N}$

绝对减值法产生的 n-gram 通常优于线性减值法。

5.3 数据平滑

◆ 四种减值法的比较：

- **Good-Turing法**：对非0事件按公式削减出现的次数，余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，余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，余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，余留出来的概率均分给0概率事件。

5.3 数据平滑

◆ 删除插值法 (Deleted Interpolation):

基本思想: 用低阶语法估计高阶语法, 即当3-gram的值不能从训练数据中准确估计时, 用2-gram来替代, 同样, 当2-gram的值不能从训练语料中准确估计时, 可以用1-gram的值来代替。插值公式:

$$p(w_3|w_1w_2) = \lambda_3 p'(w_3|w_1w_2) + \lambda_2 p'(w_3|w_2) + \lambda_1 p'(w_3) \quad (5-13)$$

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

5.3 数据平滑

◆ 删除插值法 (Deleted Interpolation):

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

将训练语料分为两部分，即从原始语料中删除一部分作为留存数据(holdout data)。

- ① 第一部分用于估计 $p'(w_3|w_1w_2), p'(w_3|w_2)$ 和 $p'(w_3)$ 。
- ② 第二部分用于计算 $\lambda_1, \lambda_2, \lambda_3$ ：使语言模型对留存数据的困惑度最小。

5.3 数据平滑

◆ 各种平滑方法的详细介绍和比较请参阅：

Chen, Stanley F. and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Model. Available from the website:

<http://www2.cs.cmu.edu/~sfc/html/publications.html>

◆ SRI 语言模型工具：

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

◆ CMU-Cambridge 语言模型工具：

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

5.4 语言模型的自适应

5.4 语言模型的自适应

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文本的类型、主题和风格等都十分敏感；
2. n 元语言模型的独立性假设的前提是一个文本中的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。

5.4 语言模型的自适应

◆ 自适应方法

- 基于缓存的语言模型 (cache-based LM)
- 基于混合方法的语言模型
- 基于最大熵的语言模型

5.4 语言模型的自适应

◆ 基于缓存的语言模型 (Cache-based LM):

针对的问题：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，比标准的 n-gram 模型预测的概率要大。

针对这种现象，cache-based自适应方法的**基本思路**是：**语言模型通过 n-gram 的线性插值得：**

$$\hat{p}(w_i | w_1^{i-1}) = \lambda \hat{p}_{Cache}(w_i | w_1^{i-1}) + (1 - \lambda) \hat{p}_{n-gram}(w_i | w_{i-n+1}^{i-1}) \quad (5-14)$$

插值系数 λ 可以通过EM算法求得。

5.4 语言模型的自适应

◆ 基于缓存的语言模型 (Cache-based LM):

处理方法: 在缓存中保留前面的 K 个单词, 每个词的概率 (缓存概率) 用其在缓存中出现的相对频率计算得出:

$$\hat{p}_{Cache}(w_i | w_1^{i-1}) = \frac{1}{K} \sum_{j=i-K}^{i-1} I_{\{w_j=w_i\}} \quad (5-15)$$

其中, I_{ε} 为指示器函数(indicator function), 如果 ε 表示的情况出现, 则 $I_{\varepsilon} = 1$, 否则, $I_{\varepsilon} = 0$ 。

5.4 语言模型的自适应

◆ 基于缓存的语言模型 (Cache-based LM):

这种方法的缺陷是，缓存中一个词的重要性独立于该词与当前词的距离。P. R. Clarkson等人 (1997) 的研究表明，缓存中每个词对当前词的影响随着与该词距离的增大呈指数级衰减，因此，将(5-15)式写成：

$$\hat{p}_{Cache}(w_i | w_1^{i-1}) = \beta \sum_{j=1}^{i-1} I_{\{w_i=w_j\}} e^{-\alpha(i-j)} \quad (5-16)$$

其中， α 为衰减率， β 为归一化常数，以使得：

$$\sum_{w_i \in V} \hat{p}_{Cache}(w_i | w_1^{i-1}) = 1, \quad V \text{ 为词汇表}$$

5.4 语言模型的自适应

◆ 基于混合方法的语言模型：

针对的问题： 由于大规模训练语料本身是异源的(heterogenous)，来自不同领域的语料无论在主题(topic)方面，还是在风格(style)方面，或者两者都有一定的差异，而测试语料一般是同源的(homogeneous)，因此，为了获得最佳性能，语言模型必须适应各种不同类型的语料对其性能的影响。

5.4 语言模型的自适应

◆ 基于混合方法的语言模型：

处理方法： 将语言模型划分成 n 个子模型 M_1, M_2, \dots, M_n ，整个语言模型的概率通过下面的线性插值公式计算得到：

$$\hat{p}(w_i | w_1^{i-1}) = \sum_{j=1}^n \lambda_j \hat{p}_{M_j}(w_i | w_1^{i-1}) \quad (5-17)$$

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

λ 值可以通过EM算法计算出来。

5.4 语言模型的自适应

◆ 基于混合方法的语言模型：

基本方法：

- (1) 对训练语料按来源、主题或类型等聚类(设为n类);
- (2) 在模型运行时识别测试语料的主题或主题的集合;
- (3) 确定适当的训练语料子集, 并利用这些语料建立特定的语言模型;
- (4) 利用针对各个语料子集的特定语言模型和线性插值公式(5-17), 获得整个语言模型。

$$\hat{p}(w_i | w_1^{i-1}) = \sum_{j=1}^n \lambda_j \hat{p}_{M_j}(w_i | w_1^{i-1}) \quad (5-17)$$

5.4 语言模型的自适应

◆ 基于混合方法的语言模型：

EM 迭代计算插值系数 λ ：

- (1) 对于 n 个类，随机初始化插值系数 λ ；
- (2) 根据公式(5-17)计算新的概率和期望；
- (3) 第 r 次迭代，第 j 个语言模型在第 $i (i \leq n)$ 类上的系数：

$$\lambda_{ij}^r = \frac{\lambda_{ij}^{r-1} p_{ij}(w|h)}{\sum_{i=1}^n \lambda_{ij}^{r-1} p_{ij}(w|h)} \quad \text{其中, } h \text{ 为历史。}$$

- (4) 不断迭代，重复步骤(2)和(3)，直至收敛。

5.4 语言模型的自适应

◆ 基于最大熵的语言模型：

基本思想：通过结合不同信息源的信息构建**一个语言模型**。每个信息源提供一组关于模型参数的约束条件，在所有满足约束的模型中，选择熵最大的模型。

5.4 语言模型的自适应

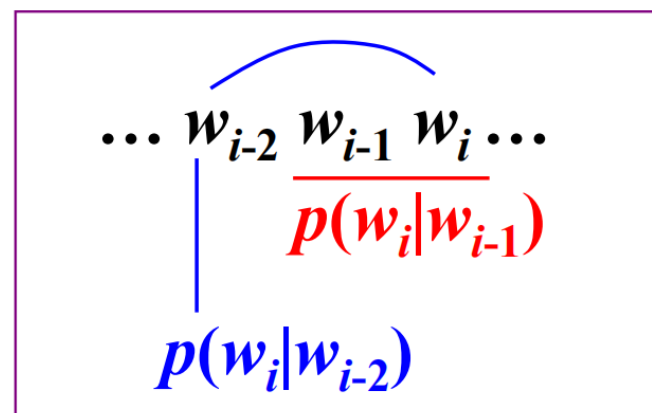
◆ 基于最大熵的语言模型：

例如：考虑两个语言模型 M_1 和 M_2 ，假设 M_1 是标准的2元模型，表示为 f 函数：

$$\hat{p}_{M_1}(w_i | w_1^{i-1}) = f(w_i, w_{i-1}) \quad (5-18)$$

M_2 是距离为2的2元模型 (distance-2 bigram)，定义为 g 函数：

$$\hat{p}_{M_2}(w_i | w_1^{i-1}) = g(w_i, w_{i-2}) \quad (5-19)$$



5.4 语言模型的自适应

◆ 基于最大熵的语言模型：

用线性插值方法通过取这两个概率估计的平均值，并采用后备(backing-off)平滑技术来解决这个问题。

最大熵原则将所有的信息源组合成一个模型，对于该模型的约束并不是让公式(5-18)和(5-19)对于所有可能的历史都成立，而是更宽松的限制，即它们在训练数据上平均成立即可，因此，公式(5-18)和(5-19)被分别改写成：

5.4 语言模型的自适应

◆ 基于最大熵的语言模型：

$$E(\hat{p}_{M_1}(w_i | w_1^{i-1}) | w_{i-1} = a) = f(w_i, a) \quad (5-20)$$

$$E(\hat{p}_{M_2}(w_i | w_1^{i-1}) | w_{i-2} = b) = g(w_i, b) \quad (5-21)$$

如果约束条件是一致的(相互之间不矛盾)，那么，总有模型满足这些条件，余下的问题就是利用通用迭代算法 (generalized iterative scaling, GIS) 选择使熵最大的模型。

5.5 语言模型应用举例

5.5 语言模型应用举例

◆ 汉语分词问题：

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 / 。

5.5 语言模型应用举例

◆ 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 $S = z_1 z_2 \cdots z_m$, $W = w_1 w_2 \cdots w_k$ ($1 \leq k \leq n$) 是一种可能的切分。那么,

$$\begin{aligned}\hat{W} &= \operatorname{argmax}_W p(W|S) \\ &= \operatorname{argmax}_W p(W)p(S|W) \\ &\cong \operatorname{argmax}_W p(W)\end{aligned}$$

最基本的做法是以词为独立的统计基元，但效果不佳。

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

具体实现时，可把汉语词汇分成如下几类：

- (1) 分词词典中规定的词；
- (2) 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。

专用名词占未登录词的95%!

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

具体进一步做如下约定，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 等作为一类；
- 对词法派生词MW和词表词LW，每个词单独作为一类。

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

➤ 例如：

3月18日下午3点比尔盖茨在北京发表讲话，决定从今年起微软亚洲研究院将大规模招收研发人员，其中，80%将从北京航空航天大学所培养的应届博士或硕士毕业生中选拔，年薪不低于3万美元。

Just a joke 😊

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

➤ 例如：

日期dat

时间tim

人名PN

地名LN

3月18日下午3点比尔盖茨在北京发表讲话，决定从
今年起微软亚洲研究院将大规模招收研发人员，其中，
80%将从北京航空航天大学所培养的应届博士或硕士毕
业生中选拔，年薪不低于3万美元。

百分数per

机构名ON

货币数mon

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

➤ 例如：

词序列变为类序列：

dat/ tim/ PN/ 在/ LN/ 发表/ 讲话/ , / 决定/ 从/
tim/ 起/ ON/ 将/ 大/ 规模/ 招收/ 研发/ 人员/ , / 其中
/ , / per/ 将/ 从/ ON/ 所/ 培养/ 的/ 应届/ 博士/ 或/
硕士/ 毕业生/ 中/ 选拔/ , / 年薪/ 不/ 低于/ mon/ 。

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

那么, $\hat{C} = \operatorname{argmax}_C p(C|S)$

$$= \operatorname{argmax}_C p(C) \times p(S|C) \quad (5-22)$$

语言模型 $\rightarrow p(C)$ $p(S|C)$ \rightarrow 生成模型

$p(C)$ 可采用三元语法：

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

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

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

生成模型在满足独立性假设的条件下，可近似为：

$$p(S|C) \approx \prod_{i=1}^N p(s_i|c_i) \quad (5-25)$$

该公式的含意是，任意一个词类 c_i 生成汉字串 s_i 的概率只与自身有关，而与其上下文无关。例如，如果“教授”是词表里的词，那么

$$p(s_i = \text{教授} | c_i = LW) = 1, \quad \text{否则 } p(s_i | c_i) = 0$$

5.5 语言模型应用举例

词 类	生成模型 $p(S C)$	语言知识来源
词表词 (LW)	若 S 是词表词, $p(S LW)=1$, 否则为0;	分词词表
词法派生词 (MW)	若 S 是派生词, $p(S MW)=1$, 否则为0;	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 S 可用实体名词规则集 G 识别, $p(S G)=1$, 否则为0。	实体名词规则集

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

➤ 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。

5.5 语言模型应用举例

◆ 采用基于语言模型的分词方法：

➤ 实验：

- (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：

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

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

汉语分词： 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/ 。

标注词性后：

这/P 篇/M 文章/N 写/V 得/D 太/D 平淡/A了/X 。/B

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

假设句子 S 是由单词串组成：

$$W = w_1 w_2 \cdots w_{\underline{n}} \quad (n \geq 1)$$

单词 $w_i (1 \leq i \leq n)$ 的词性标注为 t_i ，即句子 S 相应的词性标注符号序列可表达为：

$$T = t_1 t_2 \cdots t_{\underline{n}}$$

那么，分词与词性标注的任务就是要在 S 所对应的各种切分和标注形式中，寻找 T 和 W 的联合概率 $p(W, T)$ 为最优的词切分和标注组合。

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

(1) 基于词性的三元统计模型：

$$\begin{aligned} p(W, T) &= p(W|T) \times p(T) \\ &\approx \prod_{i=1}^n p(w_i|t_i) \times p(t_i|t_{i-1}, t_{i-2}) \end{aligned} \quad (5-26)$$

其中, $p(W|T)$ 称为生成模型, $p(w_i|t_i)$ 表示在整个标注语料中, 在词性 t_i 的条件下, 单词 w_i 出现的概率。

$p(T)$ 为基于词性的语言模型, 采用三元文法, 当 $i = 1$ 时, 取 $p(t_1)$; 当 $i = 2$ 时, 取 $p(t_2|t_1)$ 。

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法:

(2) 基于单词的三元统计模型:

$$\begin{aligned} p(W, T) &= p(T|W) \times p(W) \\ &\approx \prod_{i=1}^n p(t_i|w_i) \times p(w_i|w_{i-1}, w_{i-2}) \end{aligned} \quad (5-27)$$

其中, $p(t_i|w_i)$ 反映的是每个词对应词性符号的概率。

$p(w_i|w_{i-1}w_{i-2})$ 是普通的三元语言模型, 当 $i = 1$ 时, 取 $p(w_1)$; 当 $i = 2$ 时, 取 $p(w_2|w_1)$ 。

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

(3) 分词与词性标注一体化模型：

$$p^*(W, T) = \alpha \prod_{i=1}^n p(w_i | t_i) \times p(t_i | t_{i-1}, t_{i-2}) \\ + \beta \prod_{i=1}^n p(t_i | w_i) \times p(w_i | w_{i-1}, w_{i-2}) \quad (5-28)$$

这种综合模型的指导思想是希望通过调整参数 α 和 β 的值来确定两个子模型在整个分词与词性标注过程中所发挥作用的比重，从而获得分词与词性标注的整体最优。

(当 i 分别为 1 和 2 时的语言模型分别取一元文法概率和二元文法概率，在下面的两页中同样。)

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

分析公式 (5-27)：

$$p(W, T) = p(T|W) \times p(W) \\ \approx \prod_{i=1}^n p(t_i|w_i) \times \underline{p(w_i|w_{i-1}, w_{i-2})} \quad (5-27)$$

$p(t_i|w_i)$ 对分词无帮助，且在分词确定后会对词性标注增添偏差。

因此，实现这一模型时，可仅取公式(5-27)中的语言模型部分，舍弃词性标注部分，并令(5-28)中的 $\alpha = 1$ ，仅保留加权系数 β ，于是，(5-28)式成为：

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法:

$$\hat{p}(W, T) = \prod_{i=1}^n p(w_i | t_i) \times p(t_i | t_{i-1}, t_{i-2}) \\ + \beta \prod_{i=1}^n p(w_i | w_{i-1}, w_{i-2}) \quad (5-29)$$

对比 (5-26) 式:

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

$\hat{p}(W, T)$ 在词性标注方面无改变, 但 β 进一步增强了对分词部分的约束。这样, 分词与词性标注一体化问题转化为求解 $\hat{p}(W, T)$ 最大值的问题。

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

在确定 β 系数值时，可以根据词典中词汇 w 的个数和词性 t 的种类数目，取二者之比，即：

$$\beta = \text{词典中词}w\text{的个数} / \text{词性}t\text{的种类数}$$

在系统实现时，首先对训练文本进行预处理，将人名、地名和数字串先识别出来，然后用规定的符号分别予以替代，最后再计算相应的条件概率。

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法:

实验:

- 50,000 个常用词的词典
- 13MB 已经切分和标注好的《人民日报》语料训练 $p(w_i|t_i)$ 和 $p(t_i|t_{i-1}, t_{i-2})$
- 110MB 语料训练语言模型 $p(w_i|w_{i-1}, w_{i-2})$
- 集内测试集包含 3 个文本, 规模分别为: 1284、4265 和 9681 个词
- 集外测试集包含 4 个文本, 规模分别为: 719、4644、5627 和 13166 个词

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

测试指标：

条 件 \ 指 标		分词平均 正确率 (%)	词性标注平均正确率(%)	
			一级词性标注	二级词性标注
使用公式 (5-26)	集内	97.78	96.33	93.24
	集外	96.79	96.32	93.10
使用公式 (5-28)	集内	99.48	96.28	93.21
	集外	98.06	96.32	93.07

请参阅：高山，张艳等，基于三元统计模型的汉语分词标注一体化研究，2001年全国第六届计算语言学联合学术会议论文集， pp.116-112.

5.5 语言模型应用举例

◆ 分词与词性标注一体化方法：

说明：

- 对于汉语分词而言，不同的分词方法往往各有千秋，不要简单地从正确率高低上判断方法的好坏，正确率只从某一侧面反映了方法的性能；
- 除了汉语自动分词以外，语言模型广泛地应用于自然语言处理的各个方面，是统计自然语言处理方法中最核心、最基本的模型。

5.6 神经网络语言模型

5.6 神经网络语言模型

◆ 语言模型

- 专家语法规则模型：归纳出自然语言的语法规则
- 统计语言模型：计算一个句子的先验概率
- 神经网络语言模型：通过构建神经网络的方式来探索和建模自然语言内在的依赖关系

尽管与统计语言模型的直观性相比，神经网络的黑盒子特性决定了神经网络语言模型(Neural Network Language Model, NNLM)的可解释性较差，但这并不妨碍其成为一种非常好的概率分布建模方式。

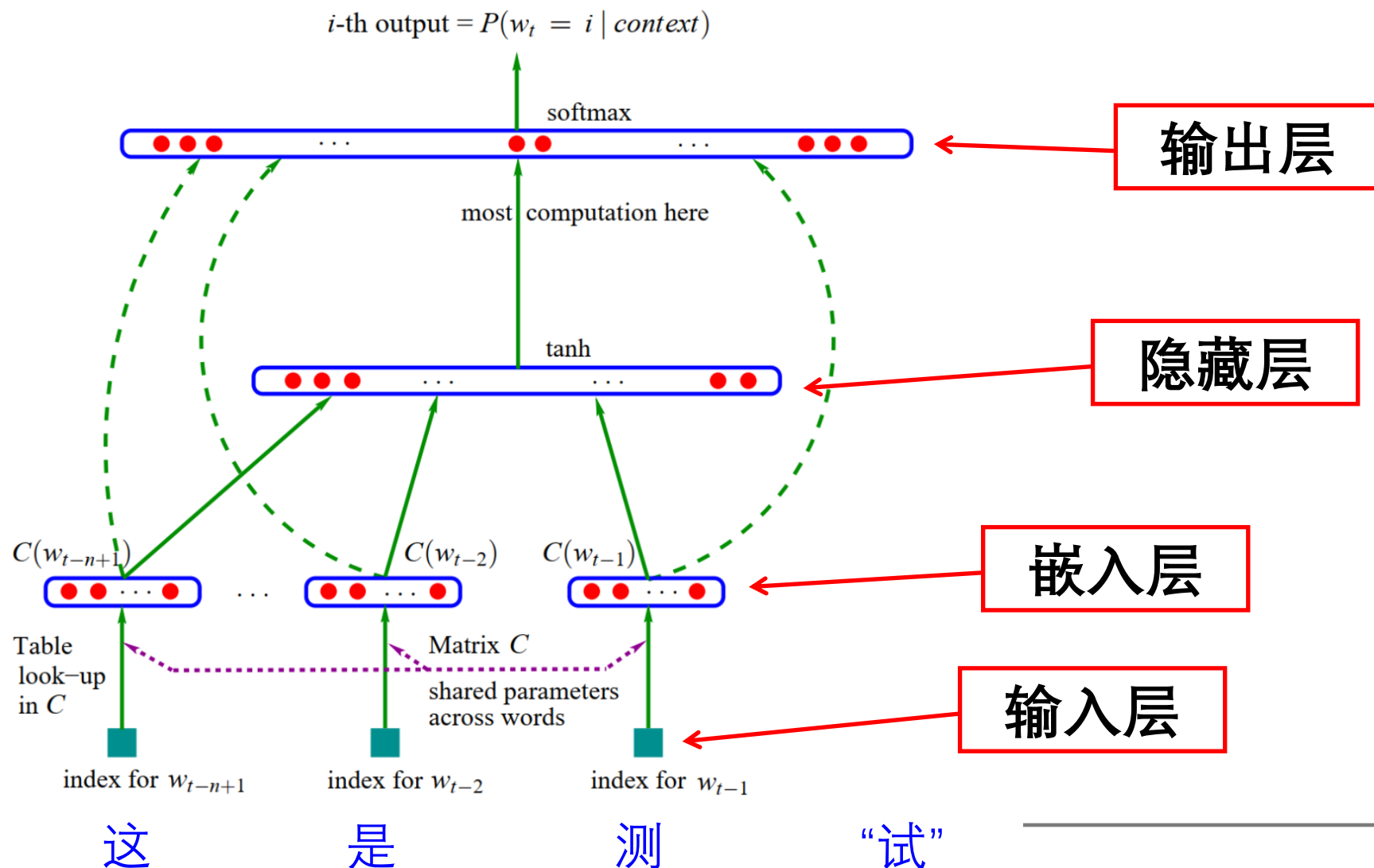
5.6 神经网络语言模型

◆ 基于前馈神经网络的模型

- 2003年Bengio提出神经网络语言模型，利用前馈神经网络建立了语言模型。
- 提出词向量的概念代替n-gram，采用连续变量(具有一定维度的实数向量)来进行单词的分布式表示，解决了维度爆炸的问题，同时通过词向量可获取词之间的相似性。

5.6 神经网络语言模型

◆ 基于前馈神经网络的模型



5.6 神经网络语言模型

◆ 基于前馈神经网络的模型

缺点：

基于前馈神经网络的模型直观上看就是使用神经网络编码的n-gram模型，也无法解决长期依赖的问题。

5.6 神经网络语言模型

◆ 基于循环神经网络的模型

在自然语言中，往往在句式相隔较远的两个词却具备一定的语法与语义关联。

例：

He doesn't have very much confidence in **himself**.

这种依赖不仅仅出现在英语中，在汉语、俄罗斯语等语言中也存在大量此类型词对组合。

5.6 神经网络语言模型

◆ 基于循环神经网络的模型

另一种长期依赖 (Long-term Dependencies) 是选择限制 (Selectional Preferences), 即选择限制主要基于已知的某人会去做某事这样的信息。

例:

1. 我要用叉子吃沙拉。
2. 我和我的朋友一起吃沙拉。

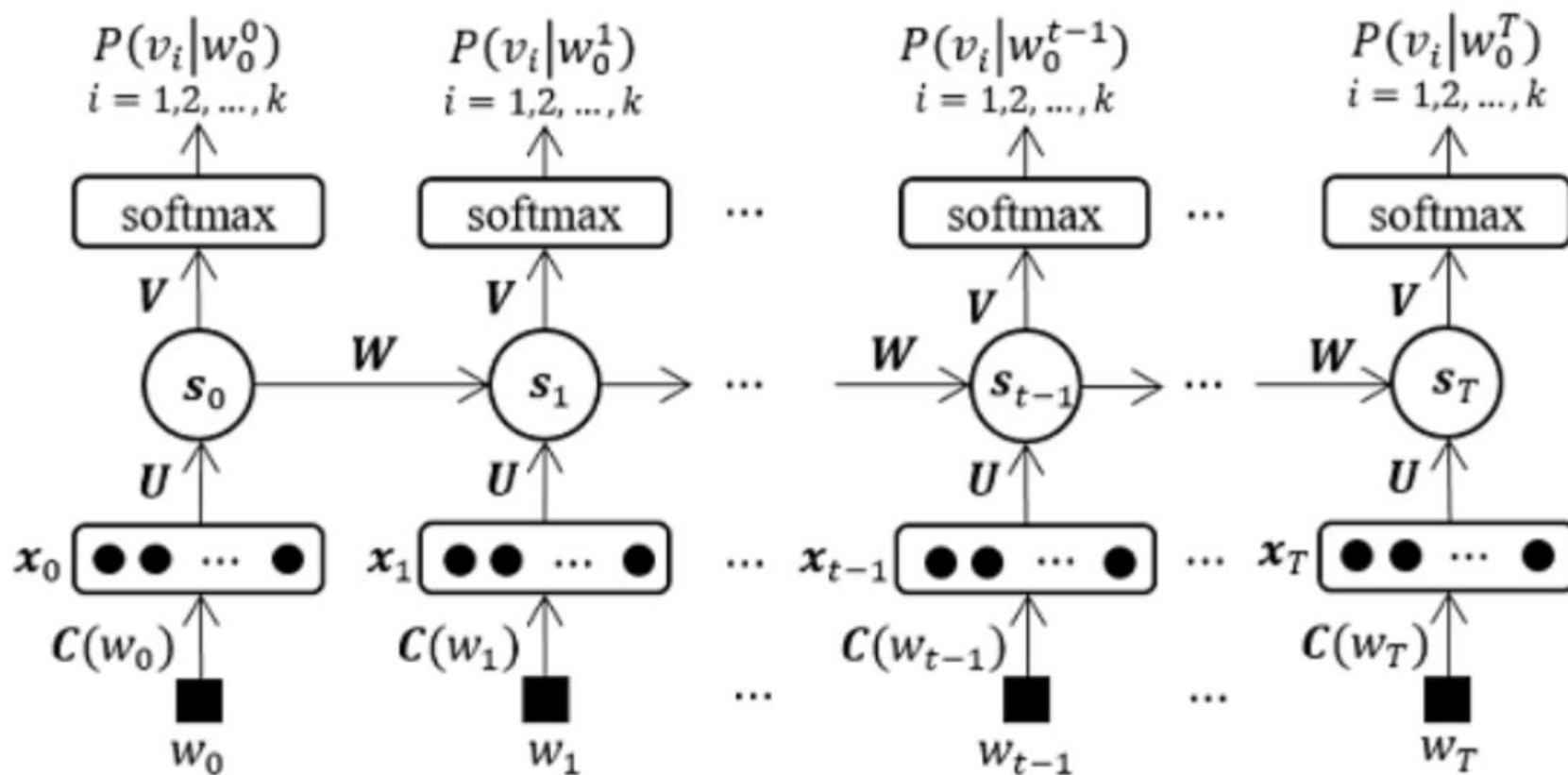
5.6 神经网络语言模型

◆ 基于循环神经网络的模型

- 为了解决定长信息的问题，Mikolov于2010年提出利用循环神经网络(Recurrent Neural Network, RNN)建立语言模型。
- 使用了RNN中的隐状态来记录词序的历史信息，能够捕获语言中的长程依赖。

5.6 神经网络语言模型

◆ 基于循环神经网络的模型



5.6 神经网络语言模型

◆ 其他基于神经网络的模型

- LSTM-RNN 语言模型
- 基于字符的 (Character-Aware) 模型
- 因式分解模型
- 双向模型 (BiRNN 和 BiLSTM)
- 注意力机制
- BERT

.....

本章小结

◆ n元语法的基本概念

◆ 数据平滑方法

- 加1法
- 减值法： 1) Good-Turing; 2) Back-off (Katz);
 3) 绝对减值(H. Ney); 4) 线性减值
- 删除减值法：低阶代替高阶

◆ 语言模型的自适应方法

Cache-based, Hybrid, ME-based

◆ 语言模型的应用

◆神经网络语言模型

习题

- 5-1. 请阅读有关文献，了解除了本讲义介绍的数据平滑方法以外的其它平滑方法；请对Good-Turing平滑方法进行简要的评价，阐述你个人的观点。
- 5-2. 利用汉语切分和标注语料（注意版权的合法性），尝试用bi-gram实现一个简单的汉语自动分词程序。
- 5-3. 阅读神经网络语言模型的相关文献，尝试实现基于RNN的语言模型。

谢谢!