第4章 *N*元文法模型

宗 成 庆

中国科学院自动化研究所

cqzong@nlpr.ia.ac.cn

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

1. 语料库概念

◆ 语料库(corpus)

—用于存放语言数据的文件(语言数据库)。

◆语料库语言学(corpus linguistics)

—基于语料库进行的语言学研究。

—研究自然语言文本的采集、存储、检索、统计、词性、句法和语义、篇章等信息的标注，以及具有上述功能的语料库在语言定量分析、词典编纂、作品风格分析和人类语言技术等领域中的应用。

1. 语料库概念

◆语料库的类型

- **按面向的任务划分**：汉语分词语料库；词性标注语料库；句法树库；面向翻译的双语平行语料库……
- **按构成划分**：同质的(homogeneous)语料库；异质(heterogeneous)语料库；系统性(systematic)语料库；专用(specialized)语料库。

◆语料库应用

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，统计方法的成功应用推动了语料库语言学的发展。

基于大规模语料库的统计方法可以

—发现语言使用的普遍规律

—通过机器学习模型自动获取语言知识

—对未知语言现象进行推测

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

2. 模型定义

- 出生于捷克一个富有的犹太家庭
- 1949 年移民美国
- 麻省理工学院获博士学位，之后在哈佛大学任教，一年后到康乃尔大学任教
- 1972年到IBM 华生实验室(IBM T.G.Watson Labs)做学术休假，领导了语音识别实验室，两年后离开康奈尔大学正式去IBM工作
- 提出了基于因马尔科夫模型的语音识别方法，**建立了*n*-gram 语言模型**，为此当选美国工程院院士
- 1990S离开IBM公司，到约翰霍普金斯大学任教，建立了世界著名的 CLSP 实验室。

—选自百度百科(https://baike.baidu.com/item/贾里尼克)

弗莱德里克.贾里尼克
(Frederek Jelinek)
(1932.11.18 ~ 2010.9.14)

2. 模型定义

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?

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

2. 模型定义

语句 $s = w_1\ w_2\ \ldots\ w_m$ 的先验概率：

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

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

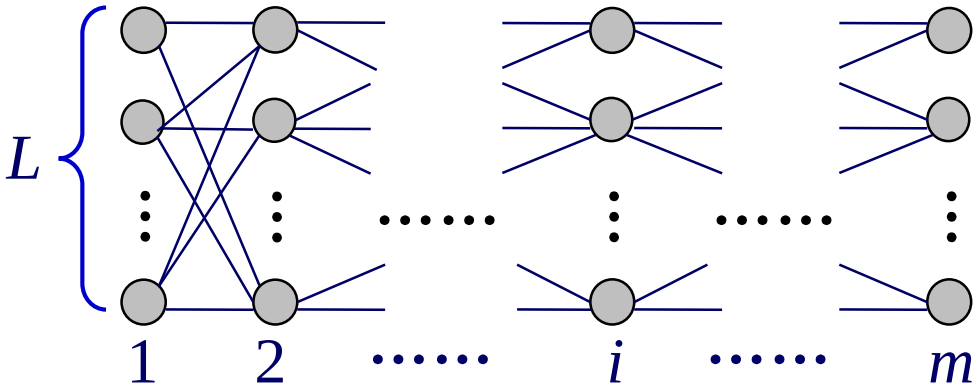
当 i=1 时，$p(w_1|w_0) = p(w_1)$。

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

2. 模型定义

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}种不同的历史组合。我们必须考虑在所有L^{i-1}种不同的历史条件下产生第i个基元的概率。那么，对于长度为m的句子，模型中有L^m 个自由参数 $p(w_m|w_1 \ldots w_{m-1})$。

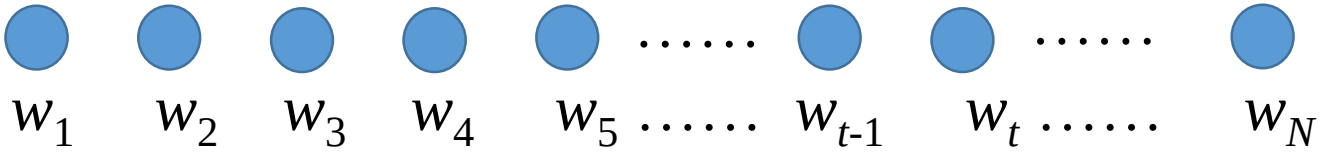
如果 L=6763, m=3, 自由参数的数目为 3.09×10^{11}！



2. 模型定义

- 解决问题的思路

如果将每一个“词(token)”看作一个状态，那么一个句子就可以看作一个状态序列。



原来：t 时刻的状态取值为 w_t $(1 \le t \le N)$ 的概率取决于前t-1个时刻及之前的状态，但**与t 的取值无关(不动性假设)**：

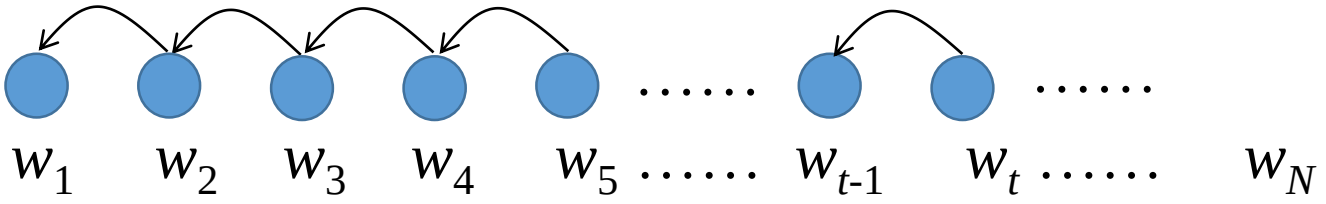
$$p(s_t = w_i \mid s_{t-1} = w_j, s_{t-2} = w_k, \cdots)$$

现在：只考虑有限个历史状态对当前“词”的影响。

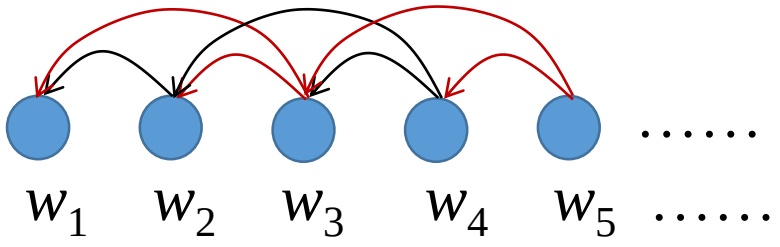
2. 模型定义

在特定情况下，如果 t时刻的状态只与其在t-1时刻的状态相关，则该系统构成一个离散的一阶马尔可夫链(First-order Markov chain)：

$$p(s_t = w_i \mid s_{t-1} = w_j, s_{t-2} = w_k, \cdots) = p(s_t = w_i \mid s_{t-1} = w_j) \quad \ldots (2)$$



如果t 时刻的状态与其在t-1 和 t-2 时刻的状态相关，则该系统构成二阶马尔可夫链(Second-order Markov chain)，等等。



以此类推，三阶、四阶…

2. 模型定义

前面n-1(n为自然数，一般为1~5)个有限的历史“词”与当前“词”构成的连续的n元词组称作n元文法(n-gram)。一个句子就是由n元构成的马尔科夫链。

- 当 n=1 时，出现在第 i 位置上的“词” w_i 独立于历史，称为一元文法，记作 uni-gram 或 monogram;
- 当 n=2 时, 构成2元文法(2-gram 或 bi-gram)，出现在第 i 位置上的“词” w_i 只与i-1位置上的基元相关。2元文法构成的序列为**一阶马尔可夫链**(First-order Markov chain);
- 当 n=3 时, 构成3元文法(3-gram 或 tri-gram)，出现在第 i 位置上的基元 w_i 与i-2 和 i-1位置上的基元相,。三元文法构成的序列为**二阶马尔可夫链**(Second-order Markov chain)。

  依次类推。

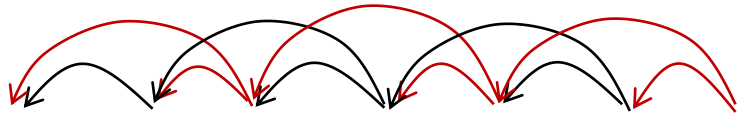
2. 模型定义

例如：

(1) I came from New York, USA and working in Beijing now.

bi-grams: I came, came from, from New, New York, York ,, , USA, USA and, and working, working in, in Beijing, Beijing now, now .

(2) 今天是个灿烂的日子。



tri-grams: 今天是个，是个灿烂，个灿烂的，灿烂的日子，的日子。

2. 模型定义

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：

$$p(s)=\prod_{i=1}^{m+1} p(w_i \mid w_{i-n+1}^{i-1}) \qquad \ldots (3)$$

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>。

这种计算语句概率的模型称为n元文法模型(n-gram model)，或称语言模型(language model, LM)。

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

3. 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。

3. 参数估计

◆实现方法

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：

$$p(w_i \mid w_{i-n+1}^{i-1}) = f(w_i \mid w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i})}{\sum_{w_i} c(w_{i-n+1}^{i})} \quad \ldots(4)$$

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 出现的相对频次。

分子为 w_{i-n+1}^{i-1} 与 w_i 同现的次数。

3. 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 元文法模型求句子的概率。

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

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

3. 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>

3. 参数估计

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

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

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

数据匮乏/稀疏
(sparse data)

数据平滑
(data smoothing)

于是，$p(Cher\ read\ a\ book) = 0$

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

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

4. 数据平滑

◆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。

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

4. 数据平滑

- 回顾—困惑度：

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：$(s_1, s_2, \cdots, s_{l_T})$，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：$PP_P(T) = 2^{H_p(T)}$

n-gram对于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。

4. 数据平滑

◆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)

见本章附录。

4. 数据平滑

(1) 加1法

- **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

- **一般性描述**：

  对于2-gram 有：

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

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

其中, V 为被考虑语料的词汇表(全部可能为历史基元)。

4. 数据平滑

在前面 3 个句子的例子中,

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

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

原来:

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

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

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

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

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

加1平滑以后:

$p(Cher|<BOS>) = ?$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

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

4. 数据平滑

类似地：

$p(read|Cher) = (0+1)/(13+1) = 1/14$

$p(a|read)=(1+2)/(13+3)=3/16,$

$p(book|a) = (1+1)/(13+2) = 2/15$

$p(<EOS>|book)= (1+1)/(13+2) = 2/15$

词汇量：$|V|=13$

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

$$p(Cher\ read\ a\ book) = \frac{1}{16}\times\frac{1}{14}\times\frac{3}{16}\times\frac{2}{15}\times\frac{2}{15} \approx 1.49\times10^{-5}$$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

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

4. 数据平滑

同理，其它 bi-grams 的概率变为：

$p(John|<BOS>) = 2/16,$

$p(read|John) = 2/14,$

$p(a|read) = 3/16,$

$p(book|a) = 2/15,$

$p(<EOS>|book) = 2/15$

<BOS>John read Moby Dick<EOS>

<BOS>Mary read a different book<EOS>

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

于是，

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

$$= \frac{2}{16} \times \frac{2}{14} \times \frac{3}{16} \times \frac{2}{15} \times \frac{2}{15}$$

$$\approx 5.95 \times 10^{-5}$$

平滑前为0.06。

4. 数据平滑

◆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

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

5. N元文法模型应用

◆以汉语分词为例

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/ 。

●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，

$$\hat{W} = \arg\max_W p(W \mid S) = \arg\max_W \left[\frac{p(W) \times p(S \mid W)}{p(S)} \right]$$

$$\simeq \arg\max_W \left[p(W) \times p(S \mid W) \right] \qquad \ldots (5)$$

5. N元文法模型应用

- 一种改进的实现方法

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. N元文法模型应用

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 $C = c_1 c_2 \ldots c_N$，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. N元文法模型应用

例如，原始句子：

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。

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/ 。

用词类符号替换后：

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/ ON/ 普外科/ 护士/ PN/ 。

5. N元文法模型应用

那么，根据(4-9)式：

$$\hat{W} = \arg\max_{W} p(W \mid S) \simeq \arg\max_{W} p(W) \times p(S \mid W) \quad \ldots (5)$$

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

语言模型　　生成模型

$p(C)$可采用trigram计算：

$$p(C) = p(c_1) \times p(c_2 \mid c_1) \prod_{i=3}^{N} p(c_i \mid c_{i-2} c_{i-1}) \quad \ldots (6)$$

$$p(c_i \mid c_{i-2} c_{i-1}) = \frac{count(c_{i-2} c_{i-1} c_i)}{count(c_{i-2} c_{i-1} \bullet)} \quad \ldots (7)$$

5. N元文法模型应用

用词类符号替换后：

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/ ON/ 普外科/ 护士/ PN/ 。

类别序列的三元文法概率：

$$p(C) = p(\text{dat1}) \times p(\text{dat2}|\text{dat1}) \times p(\text{tim1}|\text{dat1, dat2}) \times p(\text{tim2}|\text{dat2, tim1})$$
$$\times p(\text{，}|\text{tim1, tim2}) \times p(\text{typ}|\text{tim2, ，}) \times p(\text{列车}|\text{，, typ})$$
$$\times p(\text{进入}|\text{typ, 列车}) \times p(\text{LN1}|\text{列车, 进入}) \ldots\ldots$$

5. N元文法模型应用

生成模型在满足独立性假设的条件下，可近似为：

$$p(S \mid C) \approx \prod_{i=1}^{N} p(w_i \mid c_i) \quad \ldots (8)$$

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. N元文法模型应用

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. N元文法模型应用

- 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(7)；

(3)用得到的模型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. N元文法模型应用

例如，给定测试集：

(1) 10月10号张阿三在国科大介绍的文章太平淡了。

(2) 他用的手机号是13688888888，高高兴兴地给我打了个电话。

……

用初切分工具切分后：

(1-1) 10月/ 10号/ 张/ 阿三/ 在/ 国科大/ 介绍/ 的/ 文章/ 太平/ 淡/ 了/。

(1-2) 10月/ 10号/ 张/ 阿三/ 在/ 国科大/ 介绍/ 的/ 文章/ 太/ 平淡/ 了/。

(2) 他/ 用/ 的/ 手机/ 号/ 是/ 13688888888/，/ 高高兴兴/ 地/ 给/ 我/ 打/ 了/ 个/ 电话/。

……

5. N元文法模型应用

用约定符号替换：

(1-1) dat1/ dat2/ PN/ 在/ ON/ 介绍/ 的/ 文章/ 太平/ 淡/ 了/。

(1-2) dat1/ dat2/ PN/ 在/ ON/ 介绍/ 的/ 文章/ 太/ 平淡/ 了/。

(2) 他/ 用/ 的/ 手机/ 号/ 是/ FT/，/ MW/ 地/ 给/ 我/ 打/ 了/ 个/ 电话/。

……

对于候选切分(1-1)，词类序列的3元文法语言模型:

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

5. N元文法模型应用

对于候选切分(1-1)的生成模型：

(1-1) dat1/ dat2/ PN/ 在/ ON/ 介绍/ 的/ 文章/ 太平/ 淡/ 了/。

张阿三

国科大

$$p(S \mid C) \approx \prod_{i=1}^{N} p(w_i \mid c_i)$$

$$\hat{W} = \arg\max_{W} \left[p(W) \times p(S \mid W) \right]$$

5. N元文法模型应用

- 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：

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

黄昌宁，高剑峰，李沐，对自动分词的反思，见：2003年全国第七届计算语言学联合学术会议论文集，pp. 26-38

5. N元文法模型应用

- ***n*-元文法模型被广泛应用**
  - 语音识别 $p(\text{chars} \mid \text{speech})$
  - 拼音输入法 $p(\text{chars} \mid \text{pinyin})$
  - 汉语分词 $p(\text{words} \mid \text{chairs})$
  - 机器翻译 $p(\text{zh} \mid \text{en})$
  - ……

已知前面几个词就可预测出下一个最有可能出现的词：

2023年 阳春 …… ？

context $p(\text{word}|\text{context})$ 三月/白雪/面……

2023年 阳春 三月 春光明媚，……

语言生成/自动写作
(language generation /automatic writing)

生成式语言模型
(generative language model)

人工智能生成内容
(Artificial Intelligence Generated Content, AIGC)

本章小结

- N元文法模型的基本概念

  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元模型应用举例

  汉语自动分词

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

6. 习题

1. 请阅读 [Chen and Goodman, 1998] 关于数据平滑方法的技术报告，了解除了本课件介绍的数据平滑方法以外的其它平滑方法。
2. 编程实现*n*元文法模型，利用北京大学开放的《人民日报》1998年1月份的词语切分和词性标注语料，分别计算以词为基元和以词性为基元的 tri-gram 概率，分析发现可能存在的零概率文法。
3. 在上面题目的基础上，用 tri-gram 实现一种简单的汉语自动分词方法，采用网络新闻（如新浪网首页）语料测试分析你所实现的词语切分方法的性能。 然后，采用不同的数据平滑方法调试模型参数，对比不同平滑方法对分词性能的影响。

本章内容

1. 语料库概念
2. 模型定义
3. 参数估计
4. 数据平滑
5. N元文法模型应用
6. 习题
7. 附录

7. 附录

(1) 其它数据平滑方法

(2) 模型自适应

(3) 其它应用举例

7(1) 附录：其它数据平滑方法

(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概率。

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)

7(1) 附录：其它数据平滑方法

① Good-Turing 估计法

I. J. Good 于1953 年引用 Turing 方法来估计概率分布。

基本思路：假设N是原来训练语料中某个n-gram的总规模，n_r是在样本中恰好出现r次的n-gram数，即出现1次的n-gram有n_1个，出现2次的n-gram有n_2个，依次类推，出现 r 次的有 n_r 个。

例如，以“read”为历史的所有bi-gram记作：“read •”。其中，出现1次的有2053个：read cat, read dog, read fox …；出现2次的有458个：read you, read him, read her …；出现3次的有191个：read apple, read banana, read orange …… 那么，以 read 为历史的所有 bi-gram 总数为：$N = 1\times2053 + 2\times458 + 3\times191 + \ldots\ldots$

7(1) 附录：其它数据平滑方法

也就是说，
$$N = \sum_{r=1}^{\infty} n_r r \qquad \cdots (4\text{-}6)$$

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

那么，Good-Turing 估计在样本中出现r 次的n-gram的概率为：

$$p_r = \frac{r^*}{N} \qquad \cdots (4\text{-}7)$$

7(1) 附录：其它数据平滑方法

在实际应用中通常直接用n_{r+1}代替$E(n_{r+1})$，n_r代替$E(n_r)$。可以证明，原训练样本中所有事件的概率之和为：

$$\sum_{r>0} n_r \times p_r = 1 - \frac{n_1}{N} < 1 \qquad \ldots (4\text{-}8)$$

即有$\frac{n_1}{N}$剩余的概率量，将其均分给所有的未见事件($r = 0$)。

式子(4-8)的推导过程参见如下论文：

A. Nadas. on Turing's Formula for Word Probabilities.
In *IEEE Trans. on ASSP*-33, Dec. 1985. Pages 1414-1416.

7(1) 附录：其它数据平滑方法

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>

……

从样本中统计出不同 2-grams 出现的次数分别：

<BOS>John	15
<BOS>Mary	10
……	
read Moby	5
……	

7(1) 附录：其它数据平滑方法

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	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

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

$=2\times 458/2053$

$=0.446$

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

$=1.25$

……

$n_{r+1}=0$

7(1) 附录：其它数据平滑方法

得到 r^* 后，就可以应用公式(4-7) 计算概率：

$$p_r = \frac{r^*}{N} \quad \ldots (4\text{-}7)$$

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：

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

以 read作为开始、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。

7(1) 附录：其它数据平滑方法

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：

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

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	—

7(1) 附录：其它数据平滑方法

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。

基本思路：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。

另一种理解：对于每个计数 $r>K$ 的N元文法出现次数减值，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。

7(1) 附录：其它数据平滑方法

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：

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

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 取 r^*/r，r^* 由 Good-Turing 法预测。

7(1) 附录：其它数据平滑方法

那么，如何确定$\alpha(w_{i-1})$呢？

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

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

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

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

7(1) 附录：其它数据平滑方法

③ 绝对减值法 (Absolute discounting)

Hermann Ney 和 U. Essen 1993年提出。

基本思路：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 $R=L^n$)。

7(1) 附录：其它数据平滑方法

那么，样本出现了 r 次的事件的概率可以由如下公式估计：

$$p_r = \begin{cases} \dfrac{r-b}{N} & \text{当 } r > 0 \\ \dfrac{b(R-n_0)}{Nn_0} & \text{当 } r = 0 \end{cases} \quad \ldots (4\text{-}13)$$

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：$n_r \times r$。

7(1) 附录：其它数据平滑方法

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：

$$b \le \frac{n_1}{n_1 + 2n_2} < 1 \qquad \ldots (4\text{-}14)$$

留存数据：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。

请参阅： H. Ney and U. Essen. Estimating Small Probabilities by Leaving-one-Out. In *Proc. Eurospeech'*1993. Pages 2239-2242.

7(1) 附录：其它数据平滑方法

④ 线性减值法

基本思路：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。

$$p_r = \begin{cases} \dfrac{(1-\alpha)r}{N} & \text{当 } r > 0 \\ \dfrac{\alpha}{n_0} & \text{当 } r = 0 \end{cases} \qquad \ldots (4\text{-}15)$$

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。

7(1) 附录：其它数据平滑方法

- 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。

7(1) 附录：其它数据平滑方法

(3) 删除插值法

基本思想：用低阶文法协助估计高阶文法，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：

$$p(w_3 \mid w_1 w_2) = \lambda_3 p'(w_3 \mid w_1 w_2) + \lambda_2 p'(w_3 \mid w_2) + \lambda_1 p'(w_3) \quad \ldots (4\text{-}16)$$

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

- $\lambda_1, \lambda_2, \lambda_3$ 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 $\lambda_1, \lambda_2, \lambda_3$，其目标是在留存数据上使语言模型的困惑度最小。

7. 附录

(1) 其它数据平滑方法

(2) 模型自适应

(3) 其它应用举例

7(2)附录：模型自适应

◆问题提出

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；

②*n* 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。

7(2)附录：模型自适应

◆解决方法：模型自适应

(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

7(2)附录：模型自适应

- 基于缓存的语言模型
  - 基本思路：在文本中刚刚出现过词在后边的句子中再次出现的可能性往往比标准的*n*-gram模型预测的概率要大。因此，可以设置缓冲区(cache)，存储最近刚刚出现过的*K*个词汇，语言模型通过*n*-gram的线性插值求得：

$$\hat{p}(w_i \mid w_1^{i-1}) = \lambda \hat{p}_{Cache}(w_i \mid w_1^{i-1}) + (1-\lambda)\hat{p}_{n-gram}(w_i \mid w_{i-n+1}^{i-1}) \quad \ldots (4\text{-}17)$$

插值系数λ可以通过EM算法求得。

7(2)附录：模型自适应

缓存中(*K*个词)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：

$$\hat{p}_{Cache}(w_i \mid w_1^{i-1}) = \frac{1}{K} \sum_{j=i-K}^{i-1} I_{\{w_j = w_i\}} \qquad \ldots (4\text{-}18)$$

其中，I_ε为指示器函数(indicator function)，如果 ε 表示的情况出现，则 I_ε=1，否则，I_ε=0。

7(2)附录：模型自适应

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(4-18) 式写成：

$$\hat{p}_{Cache}(w_i \mid w_1^{i-1}) = \beta \sum_{j=1}^{i-1} I_{\{w_i = w_j\}} e^{-\alpha(i-j)} \qquad \ldots(4\text{-}19)$$

其中，α 为衰减率，β 为归一化常数，以使得:

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

7(2)附录：模型自适应

(2) 基于混合方法的语言模型

- 基本思路：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。

7(2)附录：模型自适应

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

$$\hat{p}(w_i \mid w_1^{i-1}) = \sum_{j=1}^{n} \lambda_j \hat{p}_{M_j}(w_i \mid w_1^{i-1}) \qquad \ldots(4\text{-}20)$$

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

λ值可以通过 EM 迭代算法计算出来。

7(2)附录：模型自适应

- 实现方法：

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 利用在各个语料子集上建立的语言模型和上述线性插值公式获得整个语言模型。

$$\hat{p}(w_i \mid w_1^{i-1}) = \sum_{j=1}^{n} \lambda_j \hat{p}_{\mathrm{M}_j}(w_i \mid w_1^{i-1})$$

7(2)附录：模型自适应

- EM 迭代计算插值系数λ：

a) 对于n个语料类，随机初始化插值系数λ；

b) 根据公式(4-20)计算新的概率和期望；

c) 第 r 次迭代，第 j 个语言模型在第i ($i \le n$)类上的系数：

$$\lambda_{ij}^{r} = \frac{\lambda_{ij}^{r-1} p_{ij}(w|h)}{\sum_{i=1}^{n} \lambda_{ij}^{r-1} p_{ij}(w|h)}$$

其中，h 为历史。

d) 不断迭代，重复步骤 b)和 c)，直至收敛。

$$\hat{p}(w_i \mid w_1^{i-1}) = \sum_{j=1}^{n} \lambda_j \hat{p}_{\mathrm{M}_j}(w_i \mid w_1^{i-1}) \quad \ldots(4\text{-}20)$$

7(2)附录：模型自适应

(3) 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。

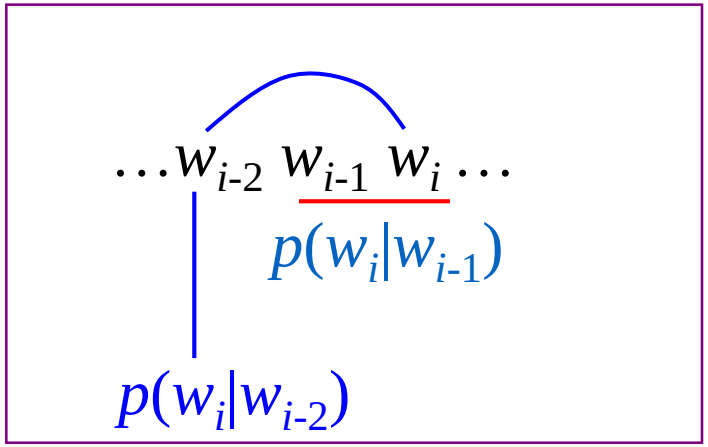
7(2)附录：模型自适应

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：

$$\hat{p}_{M_1}(w_i \mid w_1^{i-1}) = f(w_i, w_{i-1}) \quad \ldots (4\text{-}21)$$

M_2是距离为2的2元文法模型(distance-2 bigram/ skip gram)，假设将其定义为g函数：

$$\hat{p}_{M_2}(w_i \mid w_1^{i-1}) = g(w_i, w_{i-2}) \quad \ldots (4\text{-}22)$$



7(2)附录：模型自适应

用最大熵方法将这两个模型组合成一个模型时，通常无需让两个模型（公式(4-21)和(4-22)）对于所有可能的历史都成立，而是放宽约束，只要它们在训练数据上平均成立即可，于是，公式(4-21)和(4-22)被分别改写成：

$$E(\hat{p}_{M_1}(w_i \mid w_1^{i-1}) \mid w_{i-1} = a) = f_1(w_i, a) \quad \ldots (4\text{-}23)$$

$$E(\hat{p}_{M_2}(w_i \mid w_1^{i-1}) \mid w_{i-2} = b) = f_2(w_i, b) \quad \ldots (4\text{-}24)$$

根据最大熵模型：

$$p^*(w_i \mid w_{i-1}) = \frac{1}{Z(\bullet)} \exp\left(\sum_{j=1}^{l} \lambda_j \cdot f_j(w_i, \bullet)\right)$$

其中，

$$Z(\bullet) = \sum_{w_i} \exp\left(\sum_{j=1}^{l} \lambda_j \cdot f_j(w_i, \bullet)\right)$$

利用通用GIS迭代算法确定权重λ_j。

7. 附录

(1) 其它数据平滑方法

(2) 模型自适应

(3) 其它应用举例

7(3)附录：其它应用举例

◆分词与词性标注一体化方法

句子：这篇文章写得太平淡了。

切分标注后：

这/P 篇/M 文章/N 写/V 得/D 太/D 平淡/A 了/X 。/PU

这/P 篇/M 文章/N 写/V 得/D 太平/A 淡/A 了/X 。/PU

●方法描述

假设句子S 是由单词串组成：$W = w_1 w_2 \ldots w_n$ $(n \geq 1)$。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：$T = t_1\, t_2 \ldots\, t_n$。那么，分词与词性标注的任务就是要在$S$所对应的各种词语切分和词性标注形式中，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。

7(3)附录：其它应用举例

(1) 基于词性的三元统计模型

$$p(W,T) = p(W \mid T) \times p(T)$$

$$\approx \prod_{i=1}^{n} p(w_i \mid t_i) \times p(t_i \mid t_{i-1}, t_{i-2}) \qquad \ldots(4\text{-}25)$$

其中，$p(W|T)$ 称为生成模型，$p(w_i \mid t_i)$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，可采用三元文法，同前例。

7(3)附录：其它应用举例

(2) 基于词的三元统计模型

$$p(W,T) = p(T \mid W) \times p(W)$$

$$\approx \prod_{i=1}^{n} p(t_i \mid w_i) \times p(w_i \mid w_{i-1}, w_{i-2}) \qquad \ldots(4\text{-}26)$$

其中， $p(t_i \mid w_i)$ 反映的是每个词对应词性符号的概率。$p(w_i \mid w_{i\text{-}1}, w_{i\text{-}2})$ 是普通的三元语言模型, 当 i=1时，取$p(w_1)$；当 i=2 时，取 $p(w_2|w_1)$。

7(3)附录：其它应用举例

(3) 分词与词性标注一体化模型

$$p^*(W,T)=\alpha\prod_{i=1}^{n}p(w_i\mid t_i)\times p(t_i\mid t_{i-1},t_{i-2})+$$

$$\beta\prod_{i=1}^{n}p(t_i\mid w_i)\times p(w_i\mid w_{i-1},w_{i-2}) \qquad \ldots(4\text{-}27)$$

通过调整α和β 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。

分析： β 系数控制的$p(t_i\mid w_i)$对分词无帮助，且在分词确定后对词性标注又会增添偏差。因此，在实现这一模型时可以仅取公式中的语言模型部分，舍弃词性标注部分。同时令$\alpha=1$，只β控制分词比重，于是 (4-27) 式成为：

$$p^*(W,T)\approx\prod_{i=1}^{n}p(w_i\mid t_i)\times p(t_i\mid t_{i-1},t_{i-2})+\beta\prod_{i=1}^{n}p(w_i\mid w_{i-1},w_{i-2}) \qquad \ldots(4\text{-}28)$$

7(3)附录：其它应用举例

- 如何确定β系数？

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：

β=词典中词w的个数 / 词性t 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。(与前一例的做法相同。)

7(3)附录：其它应用举例

- 实验数据
  - 50,000个常用词的词典；
  - 13MB已经切分和标注好的《人民日报》语料用于训练 $p(w_i|t_i)$ 和 $p(t_i|t_{i-1}\,t_{i-2})$；
  - 110MB分词语料用于训练模型 $p(w_i|w_{i-1}w_{i-2})$；
  - 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。

7(3)附录：其它应用举例

● 测试结果

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(4-26)	集内	97.78	96.33	93.24
	集外	96.79	96.32	93.10
使用公式(4-28)	集内	99.48	96.28	93.21
	集外	98.06	96.32	93.07

高山等，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

7(3)附录：其它应用举例

◆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
- ……

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

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

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

7(3)附录：其它应用举例

$CStri$ ={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)

……

如果汉字的总数为N，使用一元文法时搜索空间为N，只选择使用频率最高的汉字；使用2元文法时搜索空间为N^2，效果比一元文法明显提高；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。

谢谢！

Thanks!