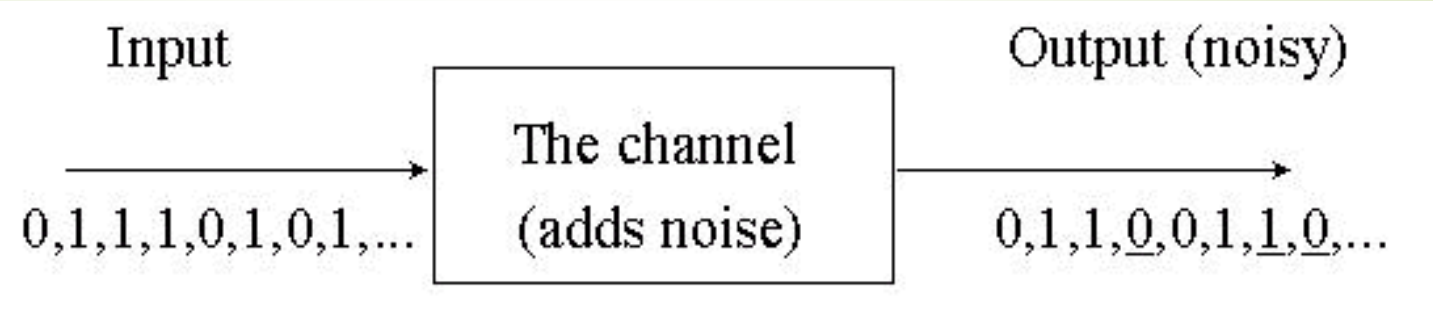
# 词处理：统计语言模型

# 主要内容

- 语言模型
- MM在汉语分词中的应用
- 基于隐马尔科夫模型的汉语分词及词性标注
- OOV识别

# 语言模型：噪声信道模型

- 噪声信道模型




- 模型：出错的概率。
- 举例：$p(0|1)=0.3, p(1|1)=0.7, p(1|0)=0.4, p(0|0)=0.6$
- 任务是：
  - 已知带有噪声的输出，想知道输入是什么。

# 语言模型：噪声信道模型的应用

- OCR
  - 文本→打印（引入噪声），扫描→图像
- 手写识别
  - 文本→神经肌肉（引入噪声），扫描→图像
- 语音识别
  - 文本→朗读（引入噪声） →声学波形
- 机器翻译
  - 目标语言→翻译（引入噪声） →源语言
- 词性标注
  - 词性序列→选择词形→文本

# 语言模型：噪声信道模型黄金规则

- 适用于OCR，手写识别，语音识别，机器翻译，词性标注等各个问题。
- 贝叶斯公式：
  - P(A|B) = P(B|A)P(A)/P(B)
- $A_{best} = argmax_A P(B|A)P(A)$
- P(B|A)是声学/图像/翻译等模型
  - 在不同领域下用不同的术语描述
- P(A)是语言模型。

# 语言模型：香农游戏

- Claude E. Shannon. “Prediction and Entropy of Printed English”, *Bell System Technical Journal* 30:50-64. 1951.
- 给定前n-1个词(或者字母),预测下一个词(字母)
- 从训练语料库中确定不同词序列概率

# 语言模型：基本概念

- 语言模型通常构建为字符串s的概率分布$p(s)$
  - 其中$p(s)$反映的是字符串s作为句子出现的概率。
  - 例如：如果一个语料库中大约100个句子中大约有1句是“Ok.”，则可以认为$p(Ok.) = 0.01$。
- 语言模型与句子是否合乎语法无关。
  - An apple ate the chicken. 可以认为其出现的概率为0。

# 语言模型：语音识别应用

- 有的词序列听起来很像，但并不是正确的句子。
- 例子1：
  - I went to a party. ✓
  - Eye went two a bar tea.
- 例子2：
  - 你现在在干什么？✓
  - 你西安载感什么？

# 语言模型：机器翻译应用

- 给定一个汉语句子，例如：王刚出现在电视上。
- 英文译文：
  - Wang Gang appeared in TV.
  - In Wang Gang appeared TV.
  - Wang Gang appeared on TV. ✓

# 语言模型：拼写检查应用

- 例子1：汉语
  - 我自己知道。✓
  - 我自已知道。
- 例子2：英语
  - Wang Gang appeared on TV.✓
  - Wang Gang appeared of TV.

# 语言模型：概率公式

- 对于由$l$个词构成的句子$s = \omega_1\omega_2 \dots \omega_l$，其概率计算公式：

$$p(s) = p(\omega_1)p(\omega_2|\omega_1)p(\omega_3|\omega_1\omega_2) \dots p(\omega_l|\omega_1 \dots \omega_{l-1})$$
$$= \prod_{i=1}^{l} p(\omega_i|\omega_1 \dots \omega_{i-1})$$

- 产生第$i(1 \le i \le l)$个词的概率是由已产生的$i-1$个词$\omega_1\omega_2 \dots \omega_{i-1}$决定的。

- $\omega_1\omega_2 \dots \omega_{i-1}$称为第$i$个词的历史。

# 语言模型：自由参数

- 随着历史长度的增加，不同的历史数目按照指数级增长。
- 若历史的长度为$i-1$，则共有$L^{i-1}$种不同的历史，其中$L$为词汇集的大小。
- 必须考虑$L^{i-1}$种不同的历史情况下，产生第$i$个词的概率。则模型中共有$L^{i}$个自由参数。
- 若$L=5000,\ i=3$,则自由参数共1250亿个。

# 语言模型：等价类映射

- 绝大多数历史不会出现在训练数据中。
- 将历史$\omega_1\omega_2 \ldots \omega_{i-1}$映射到等价类$\mathrm{E}(\omega_1\omega_2 \ldots \omega_{i-1})$，其中等价类的数目远小于全部历史的数目。
- 假设：$p(\omega_i|\omega_1 \ldots \omega_{i-1}) = p(\omega_i|\mathrm{E}(\omega_1\omega_2 \ldots \omega_{i-1}))$，则自由参数的数目会大大减少。

# n元语法(n-gram)：基本概念

- 马尔科夫假设：下一个词的出现仅依赖它前面的一个词或几个词。
- 将两个历史$\omega_{i-n+2} \dots \omega_{i-1}\omega_i$和$v_{k-n+2} \dots v_{k-1}v_k$映射到同一个等价类，当且仅当这两个历史最近的$n-1(1 \leq n \leq l)$个词相同。
- 即若$\mathrm{E}(\omega_1\omega_2 \dots \omega_{i-1}\omega_i) = \mathrm{E}(v_1v_2 \dots v_{i-1}v_i)$，则 $(\omega_{i-n+2} \dots \omega_{i-1}\omega_i) = (v_{k-n+2} \dots v_{k-1}v_k)$
- 满足上述条件的语言模型称为n元语法或n元文法。

# n元语法(n-gram)：基本概念

- 一元文法：n=1时，出现在第 $i$ 位上的词 $\omega_i$ 独立于历史，记作unigram。
- 二元文法：n=2时，出现在第$i$位上的词$\omega_i$只与前面的一个历史词$\omega_{i-1}$有关，记作bigram，也被称为一阶马尔科夫链。
- 三元文法：n=3时，出现在第$i$位上的词$\omega_i$只与与前面的两个历史词$\omega_{i-1}\omega_{i-2}$有关，记作trigram，也被称作二阶马尔科夫链。

# n元语法(n-gram)：n的选择

- 可靠性与辨别力
- 更大的n：对下一个词出现的约束性信息更多，更大的辨别力。
- 更小的n：在训练语料库中出现的次数更多，更可靠的统计结果，更高的可靠性。

# n元语法(n-gram)：bigram举例

- bigram的概率公式：$p(s) = \prod_{i=1}^{l} p(\omega_i|\omega_{i-1})$
- $p(\omega_i|\omega_{i-1})$在$i = 1$有意义，添加句首标记<BOS>。
- 所有字符串的概率之和$\sum s\, P(s) = 1$，添加句尾标记<EOS>。
- 计算概率$p(Mark\ wrote\ a\ book)$

$= p(Mark| < BOS >) \times p(wrote|Mark) \times p(a|wrote)$

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

# n元语法(n-gram)：最大似然估计

- 计算$p(\omega_i|\omega_{i-1})$，可以计算二元语法$\omega_{i-1}\omega_i$在文本中出现的频率，然后归一化。
- 用$c(\omega_{i-1}\omega_i)$表示二元语法$\omega_{i-1}\omega_i$在给定文本中的出现次数。
- 则$p(\omega_i|\omega_{i-1}) = \frac{c(\omega_{i-1}\omega_i)}{\sum_{\omega_i} c(\omega_{i-1}\omega_i)}$，称为$p(\omega_i|\omega_{i-1})$的最大似然估计(maximum likelihood estimation, MLE)。

# n元语法(n-gram)：bigram举例

- 假设训练语料S由3个句子构成：

BROWN READ HOLY BIBLE

MARK READ A TEXT BOOK

HE READ A BOOK BY DIVID

- 用计算做大似然估计的方法计算 $p(BROWM\ READ\ A\ BOOK)$。

# n元语法(n-gram)：bigram举例

- $P(BROWN \mid <BOS>) = \frac{c(<BOS>BROWN)}{\sum_{\omega} c(<BOS>\omega)} = \frac{1}{3}$
- $p(READ \mid BROWN) = \frac{c(BROWN\ READ)}{\sum_{\omega} c(BROWN\omega)} = \frac{1}{1}$
- $p(A \mid READ) = \frac{c(READ\ A)}{\sum_{\omega} c(READ\omega)} = \frac{2}{3}$

# n元语法(n-gram)：bigram举例

- $p(BOOK|\ A) = \frac{c(A\ BOOK)}{\sum_{\omega} c(A\ \omega)} = \frac{1}{2}$
- $p(< EOS > |\ BOOK) = \frac{c(BOOK<EOS>)}{\sum_{\omega} c(BOOK\omega)} = \frac{1}{2}$
- 因此，$p(BROWN\ READ\ A\ BOOK)$

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

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

$= \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$

# 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：基本思路

- P’（w）≈P(w)，但是 P’（w）≠0
- 对于一些P(w)>0, 生成P’（w）< P(w)
  - $\sum_{w \in discounted}(p(w) - p'(w)) = D$
- 分配D给所有概率为0的w： P’（w）>P(w)=0
  - 对于概率值较低的词也作调整。
- 可能某些w:P’（w）=P(w)
- 必须保证$\sum_{w \in \Omega} P'(w) = 1$
- 有许多数据平滑的方法。

# 模型评价

- 实用方法：
- 通过查看该模型在实际应用中的表现来评价统计语言模型。
  - 优点：直观，实用
  - 缺点：缺乏针对性，不够客观
- 理论方法：
- 交叉熵与困惑度（也称迷惑度，perplexity）

# 模型评价：熵

- 如果X是一个离散型随机变量，取值空间为R，其概率分布为：

$$p(x) = P(X = x), x \in R$$

那么，X的熵$H(x)$定义为：

$$H(X) = -\sum_{x \in R} p(x) log_2(x)$$

其中，约定$0 log 0 = 0$。

- 熵又称为自信息(self-information)，可以视为描述一个随机变量的不确定性的数量，它表示信源X每发一个符号所提供的平均信息量。
- 一个随机变量的熵越大，它的不确定性越大，那么，正确估计其值的可能性越小。越不确定的随机变量越需要大的信息量用以确定其值。

# 模型评价：熵

- 举例：
- 假设a, b, c, d, e, f 6个字符在某一简单的语言中随机出现，每个字符出现的概率分别为1/8, 1/4, 1/8, 1/4, 1/8, 1/8, 那么，每个字符的熵为：
- $H(p) = -\sum_{x\in\{a,b,c,d,e,f\}} P(x)logP(x)$

$$= -\left[4 \times \frac{1}{8} log \frac{1}{8} + 2 \times \frac{1}{4} log \frac{1}{4}\right] = 2\frac{1}{2}$$

- 这个结果表明，我们可以设计一种编码，传输一个字符平均只需要2.5个比特：

字符：a b c d e f

编码 :100 00 101 01 110 111

# 模型评价:相对熵（KL距离）

- 相对熵又称为Kullback-Leibler差异，或者简称为Kl距离，是衡量相同事件空间里两个概率分布相对差距的测度。两个概率分布$p(x)$和$q(x)$的相对熵定义为:

$$D(p \parallel q) = \sum_{x \in X} log \frac{p(x)}{q(x)}$$

其中约定$0\log\left(\frac{0}{q}\right) = 0, p\log\left(\frac{p}{0}\right) = \infty$

- 表示成期望值为:

$$D(p \parallel q) = E_p(log \frac{p(X)}{q(X)})$$

- 两个随机变量分布完全相同时，p=q，其相对熵为0。当两个随机分布的差别增加时，其相对熵期望也增大。

# 模型评价：交叉熵

- 交叉熵的概念是用来衡量估计模型与真实概率分布之间的差异情况的。

- 如果一个随机变量$X \sim p(x)$，$q(x)$为用于近似$p(x)$的概率分布，那么，随机变量X和模型q之间的交叉熵定义为：

$$H(X,q) = H(x) + D(p \parallel q)$$

$$= -\sum_x p(x) \log q(x)$$

$$= E_p(log \frac{1}{q(x)})$$

# 模型评价：交叉熵

- 可以定义语言$L = (X) \sim p(x)$与其模型q的交叉熵为：

$$H(L,q) = -\lim_{n\to\infty}\frac{1}{n}\sum_{x_1^n} p(x_1^n) log q(x_1^n)$$

其中，$x_1^n = x_1, x_2, \ldots, x_n$为$L$的词序列（样本），这里的词指样本中出现的任意符号单位，包括词汇、数字、标点等。$p(x_1^n)$为$x_1^n$的概率（理论值），$q(x_1^n)$为模型q对于$x_1^n$的概率估计值。

- 至此，仍然无法计算这个语言的交叉熵，因为不知道真实概率$p(x_1^n)$，不过可以假设这种语言是理想的，n趋于无穷大时，其全部单词的概率和为1。

# 模型评价：交叉熵

- 根据信息论的定理：假定语言L是稳态(stationary)遍历的(ergodic)随机过程，L与其模型q的交叉熵计算公式就变为：

$$H(L,q) = -\lim_{n\to\infty}\frac{1}{n}logq(x_1^n)$$

- 由此，可以根据模型q和一个含有大量数据L的样本计算交叉熵。在设计模型q时，目的是使交叉熵最小，从而使模型最接近真实饿概率分布$p(x)$。一般，在n足够大时（记为N），我们近似采用如下计算方法：

$$H(L,q) \approx -\frac{1}{N}logq(x_1^N)$$

# 模型评价：困惑度

- 在设计语言模型时，我们通常用困惑度(perplexity)来代替交叉熵衡量语言模型的好坏，给定语言L的样本$l_1^n = l_1, l_2, \ldots, l_n$，L的困惑度$PP_q$定义为：

$$PP_q = 2^{H(L,q)} \approx 2^{-\frac{1}{n}\log(l_1^n)} = [q(l_1^n)]^{-\frac{1}{n}}$$

- 语言模型设计的任务就是寻找困惑度最小的模型，使其最接近真实语言的情况。
- 在自然语言处理中，我们所说的语言模型的困惑度通常是指语言模型对于测试数据的困惑度。

# 课后阅读

- 统计语言模型工具及数据集：
  - http://www.52nlp.cn/language-model-training-tools-srilm-details
- 数据平滑算法
- 课本中的相关章节

# 马尔科夫(Markov)模型：概述

- 马尔科夫模型是一种统计模型，广泛的应用在语音识别，磁性自动标注，音字转换，概率文法等各个自然语言处理的应用领域。
- Markov(1856~1922)，苏联数学家。切比雪夫的学生。在概率论、数论、函数逼近论和微分方程等方面卓有成就。
- 经过长期发展，尤其是在语音识别中的成功应用，使它成为一种通用的统计工具。
- N元语言模型，是Markov模型的应用。

# 马尔科夫(Markov)模型：概述

- 随机过程又称为随机函数，是随时间随机变化的过程。马尔科夫模型描述了一类重要随机过程。
- 一个系统有N个有限状态$S = \{s_1, s_2, \dots, s_N\}$，随时间推移，系统将由某一状态转移到另一状态。
- $Q = (q_1, q_2, \dots, q_T)$为随机变量序列，其取值为状态集S中的某个状态，在时间t的状态为$q_t$。

# 马尔科夫(Markov)模型：概述

- 系统在时间t处于状态$s_j$的概率取决于其在时间1, 2, …t-1的状态，该概率为：

$$P(q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k, \dots)$$

- 离散的一阶马尔科夫链：系统在时间t的状态只与时间t-1的状态有关。

$$P(q_t = s_j | q_{t-1} = s_i, q_{t-2} = s_k, \dots) = P(q_t = s_j | q_{t-1} = s_i)$$

# 马尔科夫(Markov)模型：概述

- 马尔科夫模型：只考虑独立于时间t的随机过程

$$P(q_t = s_j | q_{t-1} = s_i) = a_{ij}, 1 \leq i.j \leq N$$

- 状态转移概率$a_{ij}$必须满足以下条件：
  - $a_{ij} \geq 0$
  - $\sum_{j=1}^{N} a_{ij} = 1$
- N个状态的一阶马尔科夫过程有$N^2$，可以表示成为一个状态转移矩阵。

# 马尔科夫(Markov)模型：举例

- 一段文字中名词，动词，形容词三类词性出现的情况可以由三个状态的马尔科夫模型描述：
- 状态$s_1$：名词
- 状态$s_2$：动词
- 状态$s_3$：形容词

# 马尔科夫(Markov)模型：举例

- 假设状态之间的转移矩阵如下：

$$A = [a_{ij}] = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{bmatrix} 0.3 & 0.5 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.2 & 0.4 \end{bmatrix} \end{array}$$

- 如果在该文字中某句子的第一个词为名词，那么该句子中三类词出现顺序为0=“名动形名”的概率。

# 马尔科夫(Markov)模型：举例

- $P(O|M) = P(s_1, s_2, s_3, s_1|M)$

$= \mathrm{P}(s_1) \cdot \mathrm{P}(s_2|s_1) \cdot \mathrm{P}(s_3|s_2) \cdot \mathrm{P}(s_1|s_3)$

$= 1 \times a_{12} \times a_{23} \times a_{31}$

$= 0.5 \times 0.2 \times 0.4$

$= 0.04$

# 马尔科夫(Markov)模型：有限状态机

- 马尔科夫模型可视为随机的有限状态机。
- 圆圈表示状态，状态之间的转移用带箭头的弧表示，弧上的数字为状态转移的概率。
- 初始状态用标记为start的输入箭头表示。
- 假设任何状态都可作为终止状态。
- 对每个状态来说，发出弧上的概率和为1。

# 马尔科夫(Markov)模型：有限状态机

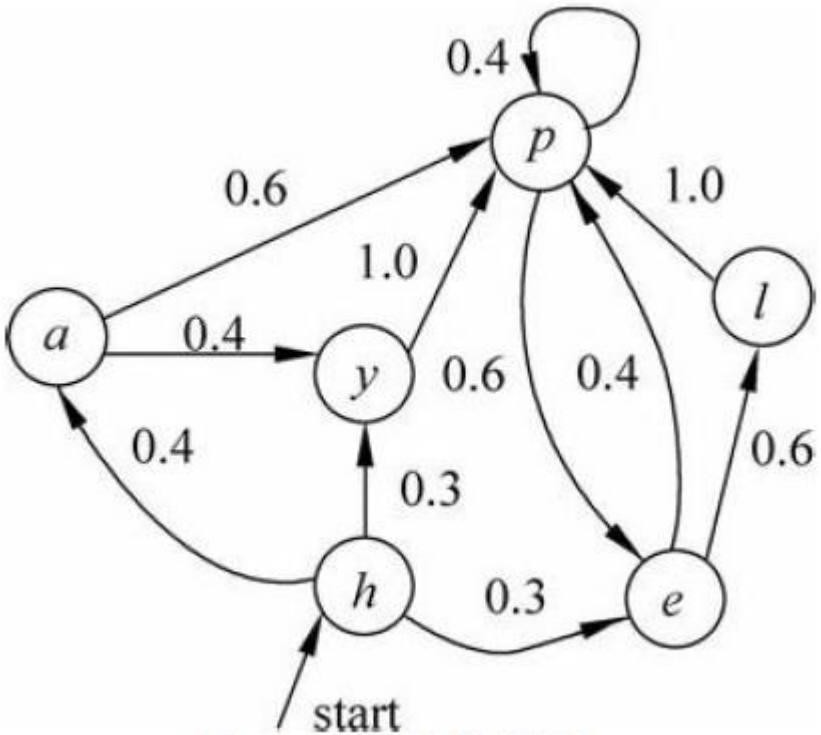



图6-4　马尔可夫模型的例子