第6章 汉字输入技术（二）

主要内容

- 汉字输入技术的发展
- 汉字键盘输入技术
- 汉字键盘输入系统
- 汉字键盘智能输入技术

6.3.1 汉字输入系统的分类

- 按系统在计算机中实现的层次分为：
  - 系统层的输入系统；
  - 应用层的输入系统。
- 按操作系统平台分为：
  - Windows下的输入系统；
  - Linux/Andriod下的输入系统；
  - Unix下的输入系统；
  - iOS下的输入系统；
  - 嵌入式系统中的输入系统。

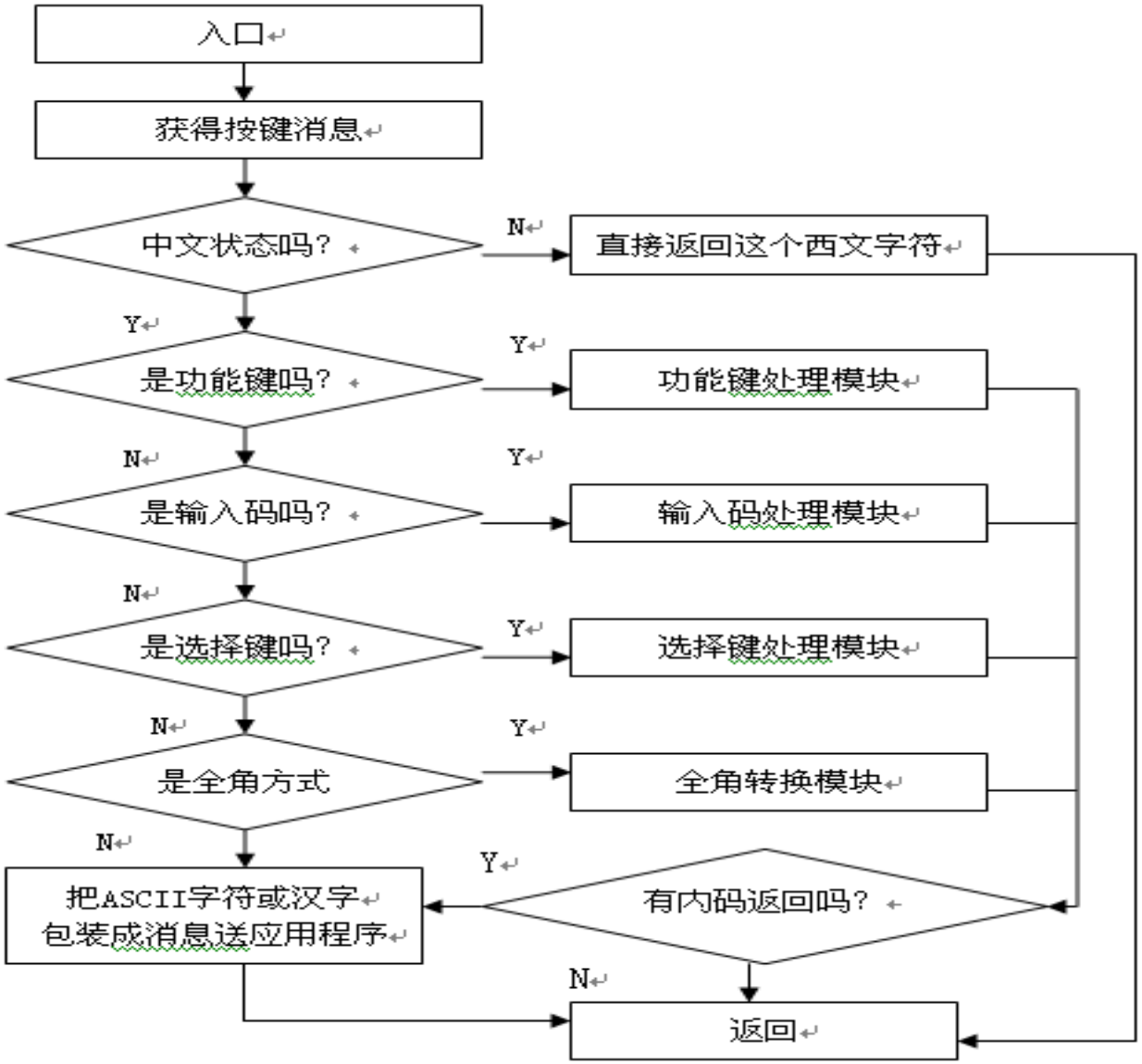
- 按支持汉字的简繁体分为：
  - 简体输入系统；
  - 繁体输入系统。
- 按支持汉字编码字符集(内码)分为：
  - GB2312的输入系统；
  - GBK的输入系统；
  - GB18030的输入系统；
  - BIG 5的输入系统；
  - ISO10646的输入系统。

- 按输入码类别分为：
  - 音码输入系统；
  - 形码输入系统；
  - 音形码输入系统；
  - 流水码输入系统；
  - 通常为综合输入系统。

6.3.2 汉字输入系统的功能

- 汉字/词组输入
- 候选字/词提示和选择
- 中英文状态切换
- 翻页
- 输入码清除、删除、结束键
- 联想
- 全角和半角

6.3.3 汉字键盘输入系统的工作流程



6.4 汉字键盘智能输入系统

- 需求：从字输入到词输入，到句输入
- 实现：统计语言模型

6.4.1 统计语言模型的产生

- 产生于基于统计方法的自然语言处理系统中。例如，语音识别、机器翻译。
- 如，对一个语音识别系统，给定语音信号a和语言的句子集合S，则系统需要解决的问题是：

$$\arg\max_{s \in S}(P|(s|a))$$

确定概率最大的句子s作为识别结果。

- 根据Bayes公式：

$$\underset{s\in S}{\operatorname{argmax}} P(s\mid a)=\underset{s\in S}{\operatorname{argmax}}\frac{P(a\mid s)P(s)}{P(a)}=\underset{s\in S}{\operatorname{argmax}} P(a\mid s)P(s)$$

由于信号波形的先验概率P(a)与s的选择无关，可以不计。
P(a|s)表示句子与信号的对应关系，也称采样模型。P(s)表示语言中句子的分布概率。

- 对于给定的句子*s*而言，通常*P*(*s*)是未知的。
- 对于一个服从某个未知概率分布*P*的语言*L*，根据给定的语言样本估计*P*的过程被称作语言建模。
- 根据语言样本估计出的概率分布*P*就称为语言*L*的语言模型。

$$\sum_{s \in L} P(s) = 1$$

- 由于构成句子的数目非常巨大，要表示所有句子的概率就空间复杂性来看是不可能的。所以，一般语言模型都是将句子的概率分布分解为各个单词的条件概率的乘积：

$$p(s) = p(w_1)p(w_2 \mid w_3)p(w_3 \mid w_1 w_2)\cdots p(w_n \mid w_1 w_2 \cdots w_{n-1})$$
$$= \prod_{i=1}^{n} p(w_i \mid w_1 \cdots w_{i-1})$$

举例：

$$P(\text{“JOHN READ A BOOK”})$$
$$=p(\text{“JOHN”})\times p(\text{“READ}|\text{JOHN”})\times$$
$$p(\text{“A}|\text{JOHN READ”})\times p(\text{“BOOK}|\text{JOHN READ A”})$$

（参数）

- 存在的问题:

1、参数空间过大，无法实用！

2、数据稀疏问题

6.4.2 N-gram模型的基本思想

- “马尔科夫假设”：下一个词的出现仅仅依赖于它前面的一 个词或者几个词.
  - 假设下一个词的出现依赖于它前面的一个词

$$P(I)=P(S)=P(w_1w_2..w_n)=p(w_1)p(w_2|w_1)p(w_3|w_1w_2)..p(w_n|w_1w_2..w_{n-1})$$

- $$\approx p(w_1)p(w_2|w_1)p(w_3|w_2)..p(w_n|w_{n-1})$$:bigram
  - 假设下一下一个词的出现依赖于它前面的两个词
  - $$\approx p(w_1)p(w_2|w_1)p(w_3|w_1w_2)..p(w_n|w_{n-2}w_{n-1})$$:trigram
  - ……

6.4.3 最大相似度估计

概率$p(w_i|w_{i-1})$一般采用最大相似度估计的方法估计（Maximum Likelihood Estimate）：

比如二元、三元概率有：

$$p(w_i \mid w_{i-1}) = \frac{c(w_{i-1}w_i)}{c(w_{i-1})} \qquad P(w_i \mid w_{i-2}w_{i-1}) \approx \frac{c(w_{i-2}w_{i-1}w_i)}{c(w_{i-2}w_{i-1})}$$

式中**c(...)** 表示一个特定词序列在整个语料库中出现的累计次数。

6.4.4 n-gram语言模型举例

- 例如，给定训练语料：
- “*John read Moby Dick*”，
  “*Mary read a different book*”，
  “*She read a book by Cher*”

求”*John read a book*”的二元文法的概率？

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

$$P(read|John) = \frac{c(John\ \ read)}{\sum_w c(John\ \ w)} = \frac{1}{1}$$

$$P(a|read) = \frac{c(read\ \ a)}{\sum_w c(read\ \ w)} = \frac{2}{3}$$

$$P(book|a) = \frac{c(a\ \ book)}{\sum_w c(a\ \ w)} = \frac{1}{2}$$

$$P(<EOS>|book) = \frac{c(book\ \ <EOS>)}{\sum_w c(book\ \ w)} = \frac{1}{2}$$

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

6.4.4 n-gram语言模型举例

- 句子的概率表现为若干bigram参数的乘积，若句子太长，计算时，会引起下溢(underflow)，采用取对数并相加的方式。

$$Ln(P(JOHN\ READ\ A\ BOOK))=$$

$$Ln(p(JOHN|<BOS>))+Ln(p(READ|JOHN))+Ln(p(A|READ))+Ln(p(BOOK|A))+Ln(p(<EOS>|BOOK))=Ln(1/3)+Ln(1)+Ln(2/3)+Ln(1/2)+Ln(1/2)=-2.8902$$

举例

假设语料库总词数为13,748词

I	3437
want	1215
to	3256
eat	938
Chinese	213
food	1506
lunch	459

举例

	I	want	to	eat	Chinese	food	lunch
I	8	1087	0	13	0	0	0
want	3	0	786	0	6	8	6
to	3	0	10	860	3	0	12
eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
food	19	0	17	0	0	0	0
lunch	4	0	0	0	0	1	0

举例

- P(I want to eat Chinese food)

=P(I)*P(want|I)*P(to|want)*P(eat|to)*P(Chinese|eat)*P(food|Chinese)

=0.25*1087/3437*786/1215*860/3256*19/938*120/213

= 0.000154171

N的选择

- 词表中词的个数 |V| = 20,000 词

n	所有可能的n-gram的个数
2 (bigrams)	400,000,000
3 (trigrams)	8,000,000,000,000
4 (4-grams)	1.6×10^{17}

6.4.5 数据稀疏问题

- 假设我们使用trigram模型

$$P(S) = p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1 w_2)...p(w_n \mid w_{n-2}w_{n-1})$$

- 如果某个 $p(w_i \mid w_{i-2}w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} = 0$
- 那么P(S)=0
- 数据稀疏问题
- 必须保证 $C \neq 0$ 从而使 $P \neq 0$

6.4.6 数据平滑技术

- MLE给训练样本中未观察到的事件赋以0概率。
- 若某n-gram在训练语料中没有出现,则该n-gram的概率必定是0。
- 解决的办法是扩大训练语料的规模。但是无论怎样扩大训练语料，都不可能保证所有的词在训练语料中均出现。
- 由于训练样本不足而导致所估计的分布不可靠的问题，称为数据稀疏问题。
- 在NLP领域中，数据稀疏问题永远存在，不太可能有一个足够大的训练语料，因为语言中的大部分词都属于低频词。

一、Zif定律

- Zipf 定律描述了词频以及词在词频表中的位置之间的关系。
- 针对某个语料库，若某个词w的词频是f，并且该词在词频表中的序号为r(即w是所统计的语料中第r常用词)，则

$$f \times r = k \text{ (k是一个常数)}$$

- 若w_i在词频表中排名50，w_j在词频表中排名150，则w_i的出现频率大约是w_j的频率的3倍。

Zif定律举例

- 例：马克吐温的小说Tom Sawyer
  共71,370 词(word tokens)
  出现了8,018 个不同的词(word types)

Word	Freq.	Use
the	3332	determiner (article)
and	2972	conjunction
a	1775	determiner
to	1725	preposition, verbal infinitive marker
of	1440	preposition
was	1161	auxiliary verb
it	1027	(personal/expletive) pronoun
in	906	preposition
that	877	complementizer, demonstrative
he	877	(personal) pronoun
I	783	(personal) pronoun
his	772	(possessive) pronoun
you	686	(personal) pronoun
Tom	679	proper noun
with	642	preposition

Tom Sawyer

Word Frequency	Frequency of Frequency
1	3993
2	1292
3	664
4	410
5	243
6	199
7	172
8	131
9	82
10	91
11-50	540
51-100	99
> 100	102

- 大部分词是低频词，3993 (50%)词(word types)仅仅出现了一次
- 常用词极为常用，前100个高频词占了整个文本的51% (word tokens)

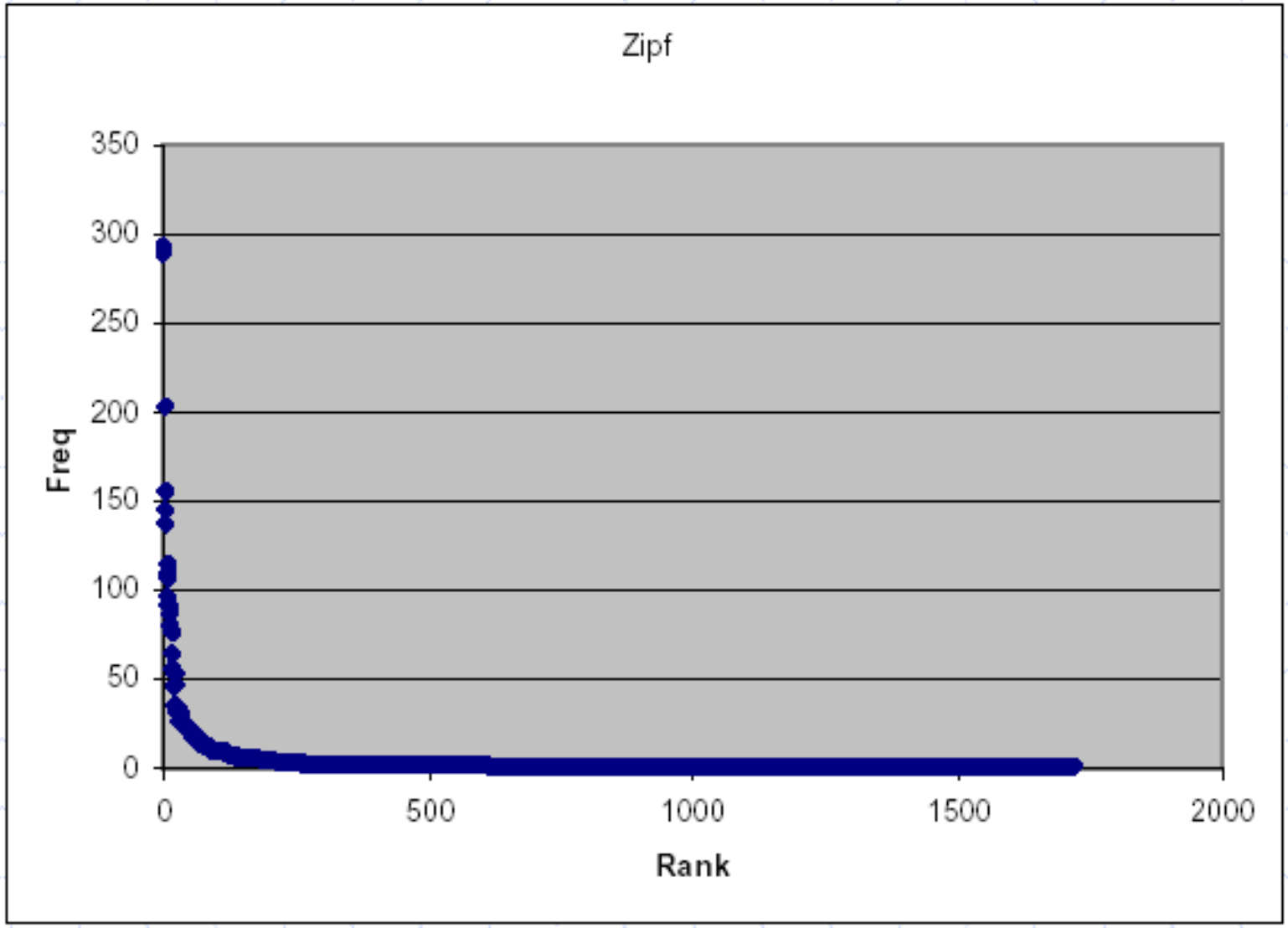
Word	Freq. (f)	Rank (r)	$f \cdot r$	Word	Freq. (f)	Rank (r)	$f \cdot r$
the	3332	1	3332	turned	51	200	10200
and	2972	2	5944	you'll	30	300	9000
a	1775	3	5235	name	21	400	8400
he	877	10	8770	comes	16	500	8000
but	410	20	8400	group	13	600	7800
be	294	30	8820	lead	11	700	7700
there	222	40	8880	friends	10	800	8000
one	172	50	8600	begin	9	900	8100
about	158	60	9480	family	8	1000	8000
more	138	70	9660	brushed	4	2000	8000
never	124	80	9920	sins	2	3000	6000
Oh	116	90	10440	Could	2	4000	8000
two	104	100	10400	Applausive	1	8000	8000

$k \approx$ 8000-9000

有例外

- 前3个最常用的词
- $r = 100$ 时

Tom Sawyer 第1-3章



- Zipf定律告诉我们语言中只有很少的常用词，语言中大部分词都是低频词(不常用的词)
- Zipf的解释是Principle of Least effort(讲话的人和听话的人都想省力的平衡)
  - 说话人只想使用少量的常用词进行交流
  - 听话人只想使用没有歧义的词(量大低频)进行交流
- Zipf 定律告诉我们对于语言中的大多数词，它们在语料中的出现是稀疏的。只有少量词语料库可以提供它们规律的可靠样本。

二、 数据稀疏问题

- “*John read Moby Dick*”,
  “*Mary read a different book*”,
  “*She read a book by Cher*”

- 考虑计算句子*CHER READ A BOOK*的概率。

  c(*CHER READ*)=0

  p(*READ*|*CHER*)＝0

  p(*CHER READ A BOOK*)=0 (有问题)

- 对语言而言，由于数据稀疏的存在，MLE 不是一种很好的参数估计办法。
- 解决办法: 平滑技术

  把在训练样本中出现过的事件的概率适当减小，把减小得到的概率密度分配给训练语料中没有出现过的事件. 这个过程有时也称为discounting(减值)。
- 减值法(Discounting)

基本思想：修改训练样本中的事件的实际计数，使样本中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。

三、 数据平滑技术

- 数据平滑技术用来对采用最大似然规则的概率估计进行调整。
- 首先它可以保证模型中任何概率均不为零。
- 其次，数据平滑使模型参数概率分布趋向更加均匀。低概率（包括零概率）被调高，高概率被调低。

6.4.7 数据平滑技术

- 加1平滑
- Good-turing平滑
- jelinek-mercer平滑
- backing-off 平滑
- katz平滑
- church-gale平滑

…

一、加1平滑

- 每一种情况出现的次数加1。
- 规定任何一个n-gram在训练语料至少出现一次（即规定没有出现过的n-gram在训练语料中出现了一次），则：new_count(n-gram) = old_count(n-gram) + 1
- 没有出现过的n-gram的概率不再是0
- 例如，对于*uni*-gram，设*w1*, *w2*, *w3* 三个词，概率分别为：1/3, 0, 2/3，加1后情况？

  2/6, 1/6, 3/6

2nd word

1st word	I	want	to	eat	Chinese	food	lunch	...	Total (N)
I	8	1087	0	13	0	0	0		3437
want	3	0	786	0	6	8	6		1215
to	3	0	10	860	3	0	12		3256
eat	0	0	2	0	19	2	52		938
Chinese	2	0	0	0	0	120	1		213
food	19	0	17	0	0	0	0		1506
lunch	4	0	0	0	0	1	0		459
...									

未平滑的 bigram 频次

	I	want	to	eat	Chinese	food	lunch	...	Total
I	.0023 (8/3437)	.32	0	.0038 (13/3437)	0	0	0		1
want	.0025	0	.65	0	.0049	.0066	.0049		1
to	.00092	0	.0031	.26	.00092	0	.0037		1
eat	0	0	.0021	0	.020	.0021	.055		1
Chinese	.0094	0	0	0	0	.56	.0047		1
food	.013	0	.011	0	0	0	0		1
lunch	.0087	0	0	0	0	.0022	0		1
...									

未平滑的 bigram 概率

平滑后的bigram 频次

频次全都加1

	I	want	to	eat	Chinese	food	lunch	...	Total (N+V)
I	~~8~~ 9	~~1087~~ 1088	1	14	1	1	1		~~3437~~ 5053
want	~~3~~ 4	1	787	1	7	9	7		2831
to	4	1	11	861	4	1	13		4872
eat	1	1	23	1	20	3	53		2554
Chinese	3	1	1	1	1	121	2		1829
food	20	1	18	1	1	1	1		3122
lunch	5	1	1	1	1	2	1		2075

	I	want	to	eat	Chinese	food	lunch	...	Total
I	.0018 (9/5053)	.22	.0002	.0028 (14/5053)	.0002	.0002	.0002		1
want	.0014	.00035	.28	.00035	.0025	.0032	.0025		1
to	.00082	.00021	.0023	.18	.00082	.00021	.0027		1
eat	.00039	.00039	.0012	.00039	.0078	.0012	.021		1
Chinese	.0016	.00055	.00055	.00055	.00055	.066	.0011		1
food	.0064	.00032	.0058	.00032	.00032	.00032	.00032		1
lunch	.0024	.00048	.00048	.00048	.00048	.0022	.00048		1

平滑后的bigram概率 $p(w_1|w_2)$

加1平滑后的N-gram模型

N: 训练语料中所有的n-gram的数量，包括重复的

V: 被考虑语料的词汇量

$$p_{add}(w_i \mid w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) + \delta}{\sum_{w_i} c(w_{i-n+1}^{i}) + \delta \mid V \mid}$$

δ =1时 对于2-gram 有：

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

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

其中，V 为被考虑语料的词汇量。

加1平滑方法的特点

- 训练语料中未出现的n-gram的概率不再为0，是一个大于0的较小的概率值。
- 但由于训练语料中未出现n-gram数量太多，平滑后，所有未出现的n-gram占据了整个概率分布中的一个很大的比例。
- 因此，在NLP中，Add-one给训练语料中没有现过的n-gram分配了太多的概率空间。
- 认为所有未出现的n-gram概率相等，这是否合理？
- 出现在训练语料中的那些n-gram，都增加同样的频度值，这是否公平？(低频、高频)

加1平滑不足举例

- AP 语料数据 (Church and Gale, 1991)
  - 语料共含有 22,000,000 个bigrams (token)
  - 语料中共出现了 273,266 个词(type) (因此共有 74,674,306,760个可能的 bigrams (type))
  - 74,671,100,000 bigrams 在语料中没有出现
  - 根据Add-one平滑，每个未出现过的 bigram 平均出现 0.000295次

74，674，306，756

Freq. from training data

Freq. from held-out data

f_{MLE}	$f_{empirical}$	$f_{add\text{-}one}$
0	0.000027	**0.000295**
1	0.448	0.000274
2	1.25	0.000411
3	2.24	0.000548
4	3.23	0.000685
5	4.21	0.000822

Add-one smoothed freq.

too high

too low

- 所有未出现过的bigram的概率分布 =
  (74,671,100,000 x 0.000295) / 22,000,000 ~**99.96** !!!!

改进的加法平滑

Add-delta 平滑：不是加1,而是加一个小于1的正数 ，δ 通常= 0.5，此时又称为Jeffreys-Perks Law或ELE 效果比Add-one好，但是仍然不理想。

$$P_{ELE}(w_1\ w_2\ \ldots w_n) = \frac{C(w_1\ w_2\ \ldots w_n) + 0.5}{N + 0.5\,V}$$

二、Good-Turing平滑

- I. J. Good 1953年引用Turing 的方法来估计概率分布。该方法的基本思想是：利用高频率*n*-gram的频率调整低频的*n*-gram的频率。
- 假设*N* 是样本数据的大小， n_r是在N元模型的训练集中正好出现*r* 次的事件的数目（在这里，事件为N元对 ）,n_r表示有多少个N元对出现了r次。

那么，$N=\sum_{r=1}^{\infty} n_r r$

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* 次的事件的概率为：

$$P_r = \frac{r^*}{N} \qquad \sum_{r>0} n_r p_r = 1-\frac{n_r}{N} < 1$$

Good-Turing平滑举例

- 建立频度-n-gram(本例为bigram)个数表(词表中词数14585，语料库中出现的各不相同的bigram总数199252个，bigram总数为617091个)。

1	138741
2	25413
3	10531
4	5997
5	3565
6	2486
7	1754
8	1342
9	1106
10	896

Good-Turing平滑举例

- 对于未出现的bigram

$$P_{GT}(w_1,..,w_n) \approx n_1/(n_0 n)$$

$$=138741/((14585*14585-199252)*\ 617091)$$

$$=1.058*10E-9$$

Good-Turing平滑举例

- 假设语料库中，某bigram 出现了1次，则

$$r^*=((r+1)n(r+1))/n(r)\approx(r+1)n2/n1$$

$$=2*25413/138741=0.3663$$

$$P=0.3663/617091=5.94E-7$$

Good-Turing平滑不足

- Good-Turing估计适合单词量大并具有大量的观察数据的情况下使用，在观察数据不足的情况下，本身出现次数就是不可靠的，利用它来估计出现次数就更不可靠了。缺乏利用低元模型对高元模型进行线性插值的思想。

三、线性插值平滑

- 线性插值平滑（Linear Interpolation Smoothing）通常也被称作Jelinek-Mercer 平滑。Jelinek 和 Mercer 在1980 年首先提出了这种数据平滑算法的思想，Brown 在1992 年给出了线性插值的平滑公式：

$$p_{\text{int}erp}(w_i \mid w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} \cdot p_{ML}(w_i \mid w_{i-n+1}^{i-1}) + (1-\lambda_{w_{i-n+1}^{i-1}}) \cdot p_{\text{int}erp}(w_i \mid w_{i-n+2}^{i-1})$$

该参数平滑技术的基本思想是利用低元n-gram 模型对高元n-gram 模型进行线性插值。用降元的方法来弥补高元的数据稀疏问题，数据估计有一定的可靠性。但是参数估计较困难。

四、 平滑的效果

- 数据平滑的效果与训练语料库的规模有关。
  - 数据平滑技术是构造高鲁棒性语言模型的重要手段。
  - 训练语料库规模越小,数据平滑的效果越显著。
  - 训练语料库规模越大,数据平滑的效果越不显著,甚至可以忽略不计。

6.4.8 典型的语言模型

- 上下文的定义决定了语言模型的不同。

$$P(w_i = w \mid C)$$

- 如果 $P(w_i = w \mid C) = P(w_i = w)$
- 这样的语言模型称为上下文无关模型。
- N元文法统计模型

$$P(w_i = w \mid C) = P(w_i = w \mid w_{i-n+1,i-1})$$

一、一元语言模型

- 采用MLE: $P(w_i = w) = \frac{N_w}{N}$ 又称为一元文法统计模型。

- 研究发现，Trigram模型仍旧是在实际应用中表现最佳的语言模型,并且成为许多其他的语言模型的重要组成部分。

二、N-pos模型

- N-pos模型（词性标注）

$$P(w_i = w \mid C) = P(w_i = w \mid g(w_{i-n+1})g(w_{i-n+2})...g(w_{i-1}))$$

或者 $P(w_i = w \mid C) = P(g(w_i) \mid g(w_{i-n+1})g(w_{i-n+2})..g(w_{i-1})) \cdot P(w_i = w \mid g(w_i))$

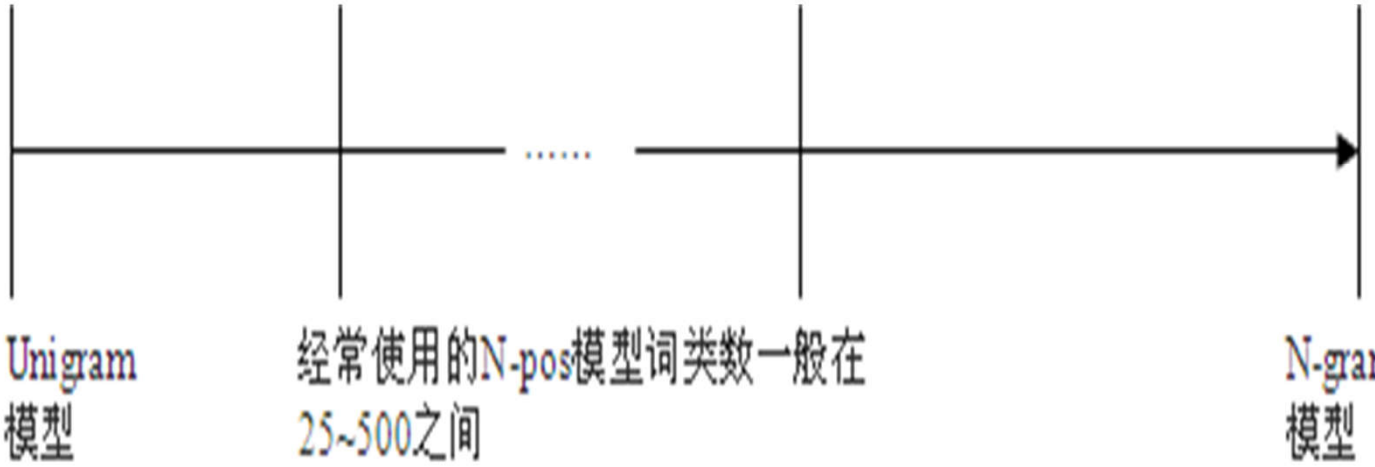
$g(w)$ 表示词w的词类

- 参数空间较小 $|G|^{N-1}|V|$，不如n-gram语言模型精确。

三、N-gram模型与N-pos模型之间的关系

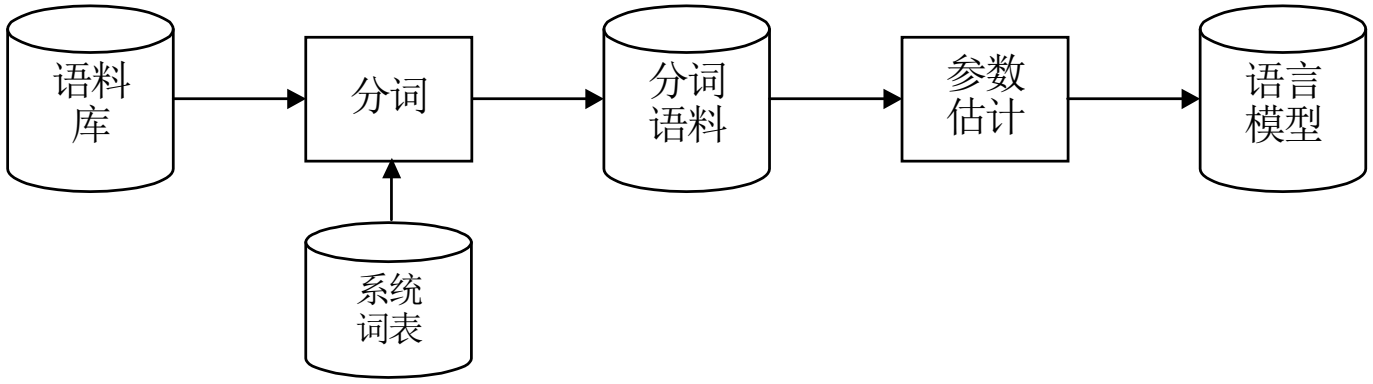
- 考察N-pos模型的极端情况,即当整个模型只有一个词类,与每一个词都有一个各自不同的词类的情况。
- 如果N-pos模型只有一个词类,那么前N-1个词类没有提供任何上下文信息,于是N-pos模型退化为Unigram模型。
- 如果每一个词都有一个各不相同的词类,那么这样的N-pos模型等价于N-gram模型。

四、N-gram模型与N-pos模型之间的关系图



6.4.9 统计语言模型参数训练的优化

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



1. 问题的分析

- 汉语统计模型的准确构建和优化涉及到以下几个问题：
  - 词表的确定（生词识别）
  - 分词
  - 参数估计
- 三者存在因果关系
  - 词表←→分词
  - 分词←→参数估计（模型）
  - 参数估计（模型）←→词表优化

2. 优化：分词-生词识别-参数估计迭代

- 非迭代过程的缺点：几个过程不能相互作用，相互促进
- 迭代过程的优点：
  - 对分词的影响：更好地解决分词歧义问题，提高分词精度
  - 对生词识别的影响：
    - 减少不合理的候选生词
    - 排除伪生词，生词自动消歧，减少后处理过程
  - 对模型的影响：逐步优化词表、提高参数估计精度、提高输入码（音）到字的转换正确率。

3. 基于N-gram信息的生词获取

- 基本思想：N元对→词频过滤→互信息过滤→校正→生词获取
- 词频(Word Frequence) --WF
- 互信息（Mutual Information）--MI

$$I(w_1; w_2) = \log \frac{p(w_1, w_2)}{p(w_1) \times p(w_2)}$$

- 词频与互信息的关系
- 候选生词的校正

4. 基于N-gram信息的生词获取

- 结果分析
  - MI因素：准确率随着互信息值的升高而升高，但抽取出的元组的数量也随之减少 ；
  - WF因素：准确率随着词频值的升高而降低，抽取出的元组数量也随之减少。
- 影响结果的其他因素
  - 分词错误：“不容\易”、“和服\务”
  - 量词：“个百分点”、“集电视连续剧”、“届世乒赛”
  - 姓氏：“新华社记者刘”、“新华社记者王”

5. 应用举例： 中文人名识别

- 基于统计和规则方法的人名识别

- 问题的提出
- 主要方法
  - 规则方法：准确；规则库冲突、复杂、庞大
  - 统计方法：资源少、效率高；准确率低
  - 混合方法：取长补短

（1）建立规则库

- 中文姓名用字特点（82年人口普查结果）
  - 729个姓氏用字
  - 姓氏分布很不均匀，但相对集中
  - 有些姓氏可用作单字词
  - 名字用字分布较姓氏要平缓、分散
  - 名字用字涉及范围广
  - 某些汉字既可用作姓氏，又可用作名字用字

（2）建立姓氏字频率库和名字用字频率库

- 姓氏用字频率库和名字用字频率库：653个单姓氏，15个复姓，1894个名字用字：

$$p(c\text{作为姓氏}) = \frac{c\text{用作姓氏的次数}}{c\text{的总出现次数}}$$

$$p(c\text{作为名字用字}) = \frac{c\text{用作名字用字的次数}}{c\text{的总出现次数}}$$

（3）建立人名常用词表

- 名字常用词表

朝阳	劲松	爱国
建国	立新	黎明
宏伟	朝晖	向阳
海燕	爱民	凤山
雪松	新民	剑峰
建军	红旗	光明

（4）建立称谓库

- 称谓库
  - 三种类型
    - 只能用于姓名之前，如：战士、歌星、演员等；
    - 只能用于姓名之后，如：阁下、之流等；
    - 姓名前后皆可，如：先生、主席、市长等。
  - 称谓前缀表：“副”、“总”、“代”、“代理”、“助理”、“常务”、“名誉”、“荣誉”等

（5）建立指界词表和标点符号表

- 简单上下文
  - 指界词表：约110个词
    - 动词：说、是、指出、认为、表示、参加等；
    - 介词：在、之、的、被 、以等；
    - 正在、今天、本人、先后等。
  - 标点符号集
    - 人名出现在句首或句尾（包括分句）的机会比较大，标点符号可用来帮助判断人名的边界。
    - 顿号一边是人名时，另一边的候选人名的可靠性高。

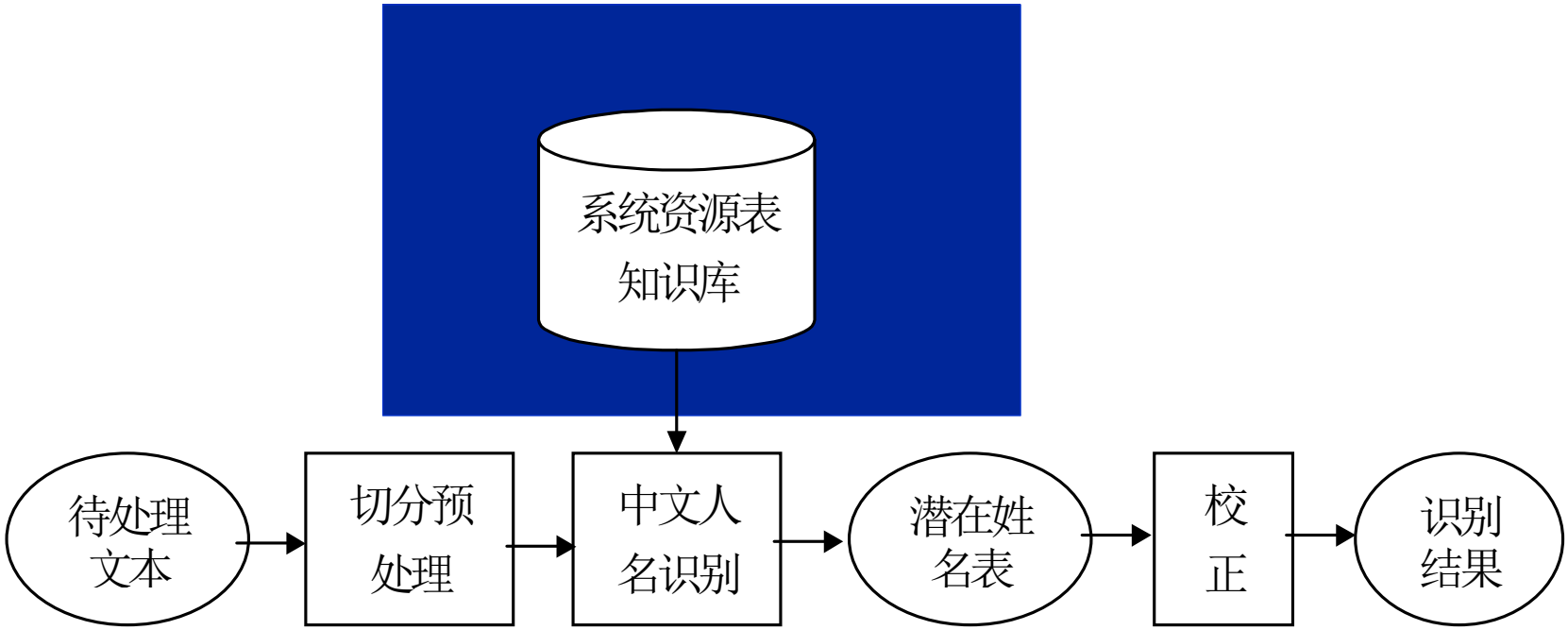
（6）建立非名字用词表

- 非名字用词表：有些双字词，如：时间、奖励、纬度等不作名字用词，但因为组成它们的单字可作为名字用字，如果跟在姓氏后面，往往会将其与可作姓氏的字一起误判为姓名。

例：

“做\这\件\事\花\了\我们\一\段\时间\。\”

（7）中文人名识别过程



（8）人名识别的具体实现

- →姓氏判别
- →名字识别
- →概率判断

候选字符串为人名的概率为：

P = 姓氏部分为姓氏的概率P1 *
余下部分的汉字作名字用字的概率P2*P3
（单名时，为P2）

(9) 校正(对潜在人名的后处理)

- 自动校正：如果两个潜在人名相似，考察它们的权值。一高一低时，将低权的潜在人名清除；都为高权时，两者都认为是人名；都是低权时，则各自通过第三个字作名字用字的概率大小来判断。概率够高，识别为人名。否则将第三个字去掉。
- 人工校正

（10）人名识别结果与分析

- 实验结果：8个测试样本，共22000多字，共有中文人名270个。系统共识别出中文人名330个，其中267个为真正人名。

  召回率=267/270*100% =98.89%

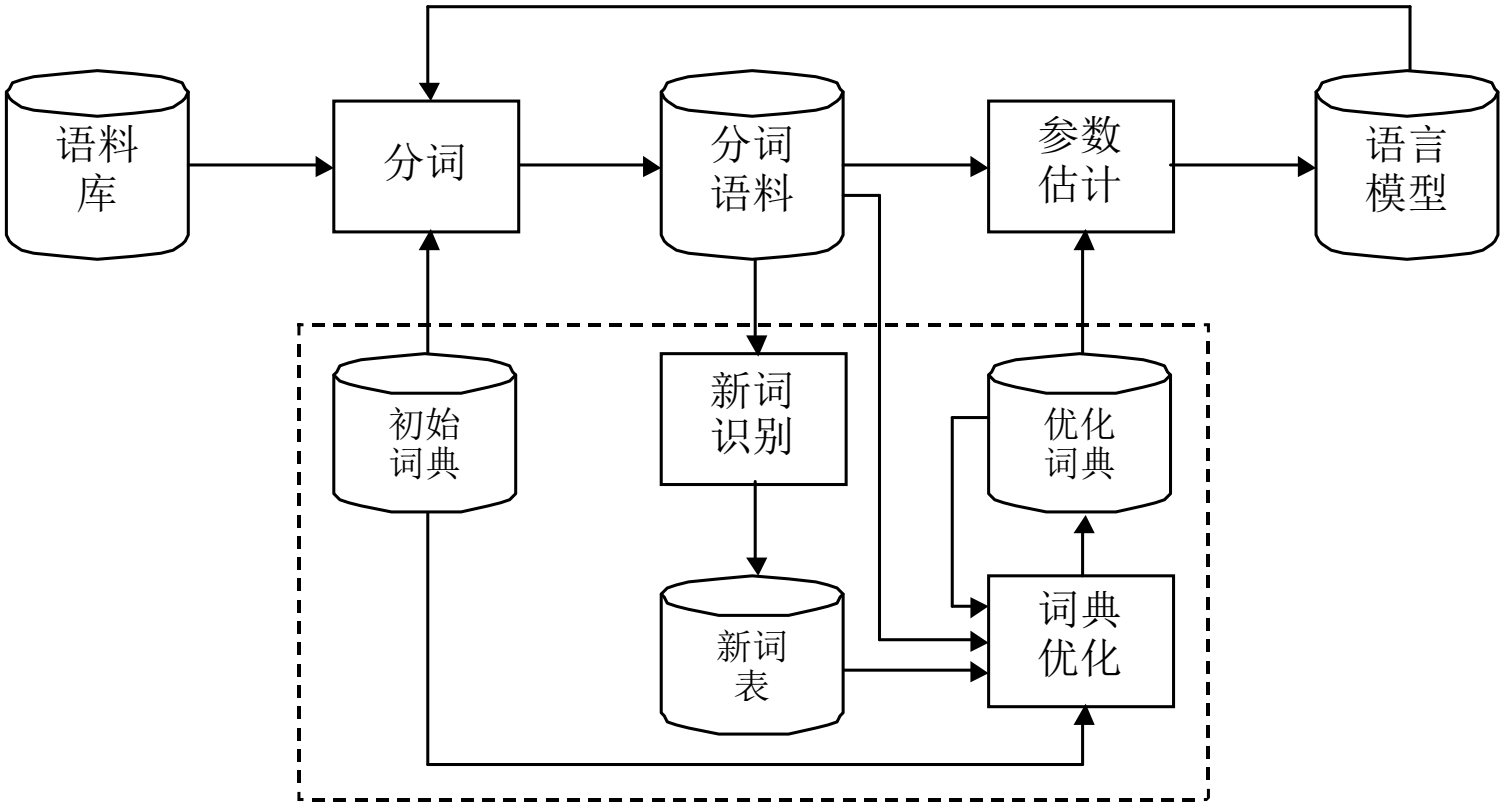
  准确率=267/330*100% =80.91%

  准确率和召回率是互相制约的，可通过概率阈值的调整来调节二者的关系。

（11）人名识别结果与分析

- 产生错误的主要原因
  - 被未识别的地名干扰。“湖北\英\山\县\詹\家\河\乡\陶\家\河\村\，\ ”
  - 受非中式人名的干扰。“司\马\义\·\艾\买\提\ ”
  - 分词结果不理想。“为\迎接\香港\回\归\送\贺\礼\”
  - 规则不准确。“南\宋\大\诗人\杨\万\里\“\惊\如\汉\殿\三\千\女\，\ ”
  - 其他。“全世界\每年\影片\产量\高\达\两\三\千\部\，\ ”

（12）N-gram模型参数训练的迭代算法



（13）实验结果和分析

- 用Bi-gram模型进行了实验，初始词典规模为24686，用2000万字的《人民日报》语料。
- 实验表明，大部分的新词在第一次迭代过程中即已识别出来，后面的迭代过程只进行较小的调整并很快趋于收敛。经过三次迭代过程，共接受新词911个，其中中文人名359个。
- 分别采用FMM方法、不带新词识别的优化语言模型和带新词识别的优化语言模型进行分词实验，通过对测试语料的抽样检查，分词准确率分别达到96.4%、97.3%和97.9%。

6.4.8动态自适应语言模型的建立

- 在自然语言中，经常出现某些在文本中通常很少出现的词，在某一局部文本中突然大量出现的情况。
- 能够根据词在局部文本中出现情况动态地调整语言模型中的概率分布数据的语言模型称为动态自适应语言模型。

建模方法

- 将N个最近出现过的词 $w_{i-N}w_{i-N+1}...w_{i-1}$ 存于一个缓存中，作为独立的训练数据
- 通过这些数据,计算动态频度分布数据

$$P_{dynamic}(w_i \mid w_{i-2}w_{i-1})$$

- 将动态频度分布数据与静态分布数据（由大规模性语料训练得到）通过线性插值的方法相结合：

$$P_{combination}(w_i \mid w_{i-2}w_{i-1}) = \lambda P_{dynamic}(w_i \mid w_{i-2}w_{i-1}) + (1-\lambda)P_{static}(w_i \mid w_{i-2}w_{i-1})$$

$$0 < \lambda < 1$$

6.4.9 统计语言模型的评价

- 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
  - 交叉熵与迷惑度

统计语言模型的评价

- 随机变量X的熵

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

统计语言模型的评价

- 长度为n的单词序列的熵

$$H(w_1, w_2, ..., w_n) = -\sum_{W_1^n \in L} p(W_1^n) \log(W_1^n)$$

- 熵率（entropy rate）

$$ER = \frac{1}{n} H(W_1^n) = -\frac{1}{n} \sum_{W_1^n \in L} p(W_1^n) \log p(W_1^n)$$

- 语言的熵

$$H(L) = \lim_{x \to \infty} \frac{1}{n} H(w_1, w_2 ... w_n) = \lim_{x \to \infty} -\frac{1}{n} \sum_{W_1^n \in L} p(W_1^n) \log p(W_1^n)$$

Kullback-Leibler (KL)距离

- Kullback-Leibler (KL)距离（相关熵）

  两个概率密度函数p(x)与q(x)它们的相关熵定义为：

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

- 描述了两个概率分布之间的差异

  （D(p||q)==0 iff p=q）

语言与其模型的交叉熵

- 某语言L，其真实的概率分布为p(x)，我们构造的该语言的概率模型为q(x)，那么该语言与其模型的交叉熵为：

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

语言与其模型的交叉熵

- 如果我们将语言视为平稳各态可遍历的随机过程：

  那么

$$H(L,m) = -\lim_{n\to\infty}\frac{1}{n}\log m(x_{1n}) \approx -\frac{1}{n}\log m(x_{1n})$$

迷惑度定义为：

$$perplexity(x_{1n},m) = 2^{H(x_{1n},m)} = m(x_{1n})^{-\frac{1}{n}}$$

即语言模型对于测试数据的困惑度。

字音转换问题

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

。 。 。 。 。 。 。

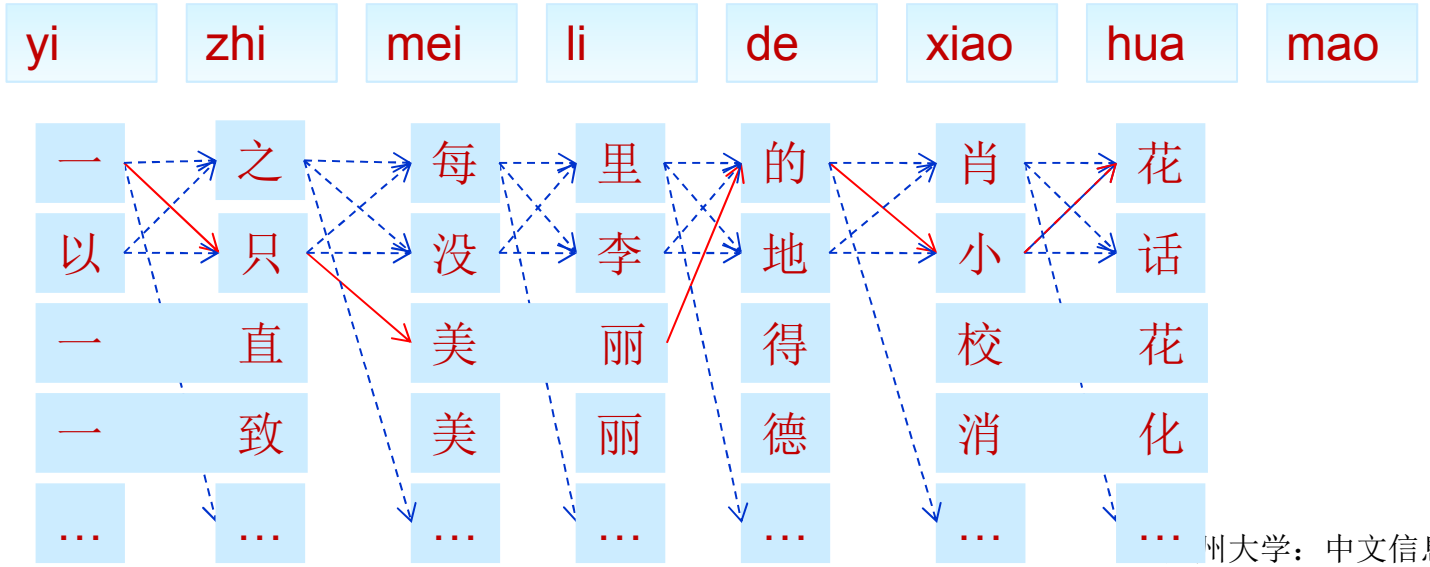
。 。 。 。 。

音字转换系统

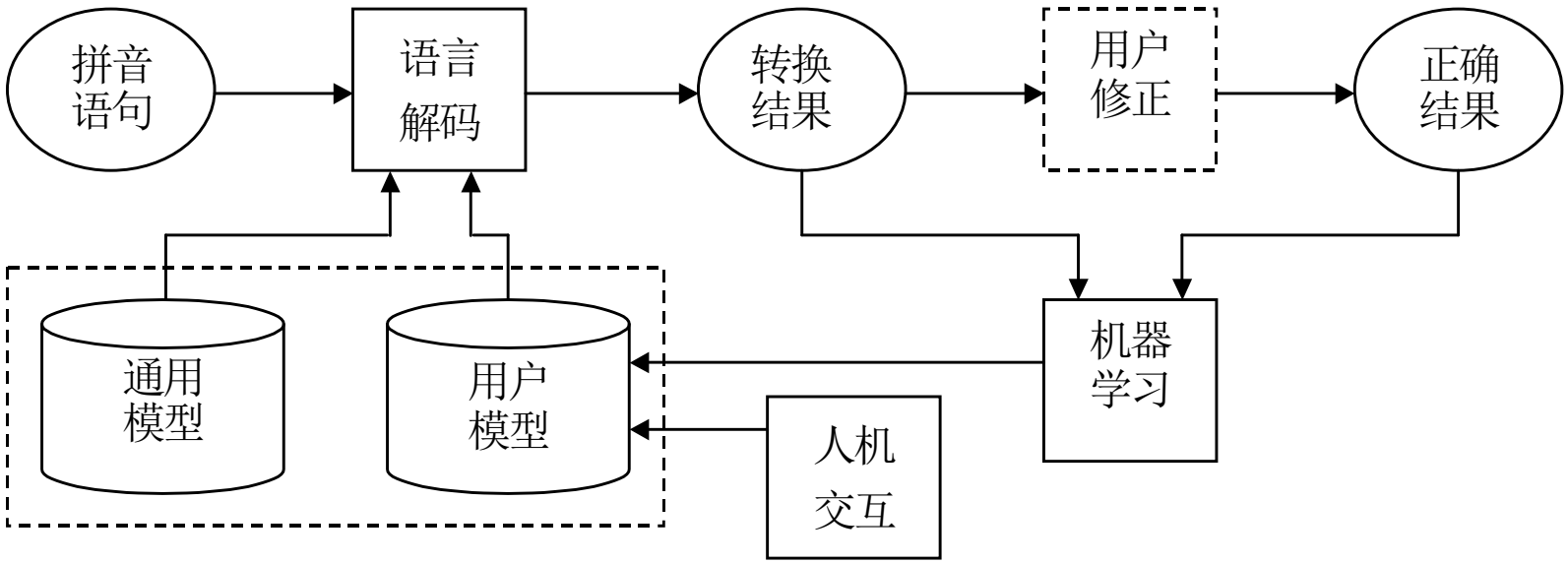
- 理论：

$$T = \arg\max_{T}(P(T \mid S)) = \arg\max_{T} P(T)P(S \mid T)$$

- 数据结构



拼音字句转换系统框架



作业

1、概述N元文法模型和N-POS模型

2、数据平滑技术的概念，概述加法平滑和Good-Turing平滑的原理。

3、给出一个智能拼音输入系统的系统框架，并说明实现动态输入的原理。