计算语言学

第 2 讲　基础知识

刘群
中国科学院计算技术研究所
liuqun@ict.ac.cn

中国科学院研究生院 2011 年春季课程讲义

内容提要

规则方法和统计方法 (1)

- 最早期的机器翻译系统主要采用了一些简单的统计方法，根据词频和搭配的概率选择译文词和调整词序
- 1970 ～ 1980 年代，随着 Chomsky 语言学和人工智能的兴起，规则方法逐渐占据了主流
- 1980 年代后期到 1990 年代初开始，统计方法重新崛起，并且在 1990 年代中期出现了一次经验主义与理性主义的论战
- 随着时间的推移，人们逐渐认识到，统计方法和规则方法并不是对立的，二者可以互为补充，缺一不可，关键问题是如何融合

规则方法和统计方法 (2)

- 经验主义和理性主义是两种不同的语言学思潮，具有相当的哲学思辨意味
  - 经验主义认为语言主要是一种经验的习得
  - 理性主义认为人的语言机制是天生的，后天的学习只是参数的调整
- 个人观点：统计方法和规则方法都是计算的手段。简单地把规则方法归结于理性主义方法，统计方法归结于经验主义方法，是把工具层面的问题上升到了哲学层面，是不合适的。

内容提要


基础知识
规则方法与统计方法
经典自然语言处理基础知识
形式语言与自动机
编译原理
数理逻辑
统计自然语言处理基础知识
简单文本处理

如何描述一种语言

- 枚举
  - 给出语言中的所有句子
  - 对于含无限多个句子的语言不合适
- 语法
  - 给出生成语言中所有句子的方法
  - 当且仅当能够用该方法产生的句子才属于该语言
- 自动机
  - 给出识别该语言中句子的机械方法

形式语法 (1)

- 形式语法：四元组 $G = \langle V_T, V_N, S, P \rangle$
- 终结符（Terminals）的有限集合 V_T
  - 终结符是句子中实际出现的符号
  - 相当于单词表（有时也称为字母表）
- 非终结符（Non-terminals）的有限集合 V_N
  - 非终结符在句子中不实际出现
  - 但在推导中起变量作用
  - 相当于语言中的语法范畴

形式语法 (2)

- 起始符 S
  - S 属于 V_N
  - 相当于句法范畴中的句子
- 重写式规则（ Rewriting Rules ）的有限集合 P
  产生式规则（ Production Rules ）的有限集合 P
  - 基本形式： $\alpha \rightarrow \beta$
  - 含义：将 α 改写成 β
  - α 和 β 是终结符和非终结符组成的串
  - α 非空， β 可以为空

形式语法 (3)

- 直接推导： $\alpha\ x\ \beta \Rightarrow \alpha\ y\ \beta$

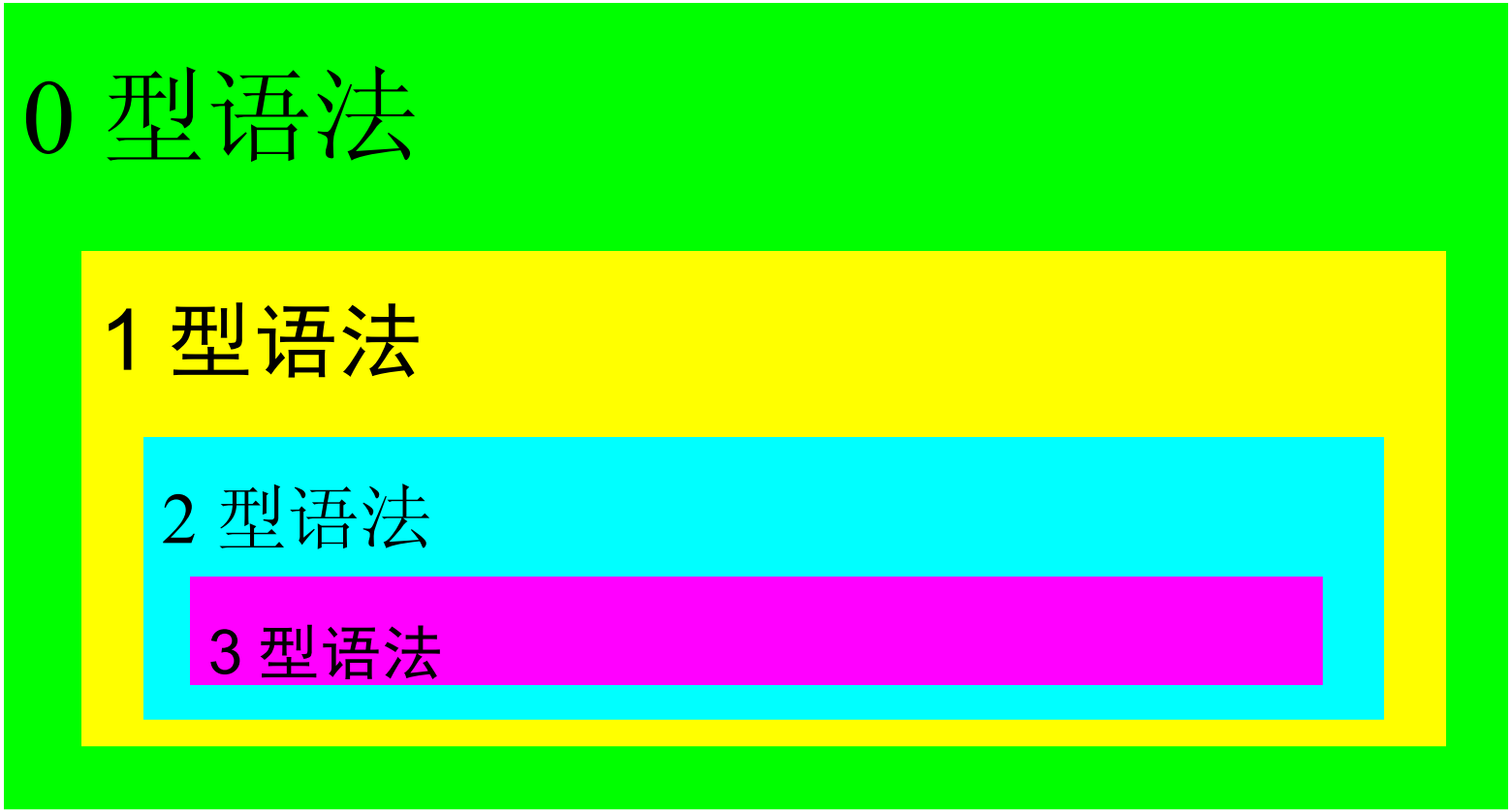
  如果 $x \rightarrow y$ 是 P 中的规则

- 推导： $\alpha \Rightarrow^* \beta$

  如果 α 可以经过多次直接推导得到 β

- 语言： $L(G)=\{\alpha \mid \alpha \in V_T^*;\ S \Rightarrow^* \alpha\}$

乔姆斯基的语法层级



乔姆斯基 0 型语法

- 短语结构语法，无限制重写语法
  PSG： Phrasal Structure Grammar
- 对规则形式的约束
  - 对于规则形式没有任何限制

乔姆斯基 1 型语法

- 上下文有关语法，上下文敏感语法
  CSG： Context Sensitive Grammar
- 对规则形式的约束：
  - $\alpha \rightarrow \beta$
    α, β 是任意串，且 α 的长度小于 β 的长度
  - $\alpha A \gamma \rightarrow \alpha \beta \gamma$
    A 是非终结符， α, β, γ 是任意串
  - 以上两种形式等价
  - 敏感：在一定的上下文环境下 A 可改写为 β

乔姆斯基 2 型语法

- 上下文无关语法，上下文自由语法
  CFG： Context Free Grammar
- 对规则形式的约束：
  - A → α： A 是非终结符， α 是任意串
  - 在任何上下文环境下 A 可改写为 α

乔姆斯基 3 型语法

- 正规语法，正则语法

  RG： Regular Grammar
- 对规则形式的约束
  - A → Bx 或者 A → x， A,B 是非终结符，x 是终结符
- 一部正则语法可以表示为一个正则表达式

  例子： {a{b|c}*}+[d|e]{f|g|h}+

乔姆斯基语法层级—例子

- P={S→A1, A→A0, A→0}
  - L(G)={$0^m1 \mid m>=1$}
  - 是正则语法
- P={S→0S1, S→01}
  - L(G)={$0^n1^n \mid n>=1$}
  - 是上下文无关语法，但不是正则语法
- P={S→0SBC, S→0BC, CB→BC,
  0B→01, 1B→11, 1C→12, 2c→22}
  - L(G) ={$0^n1^n2^n \mid n>=1$}
  - 是上下文有关语法，但不是上下文无关语法语法

乔姆斯基层级以外的语法类别

- 适度 (mild) 上下文有关语法：介于 CFG 和 CSG 之间的语法类别
  - 索引语法（ IG: Index Grammar ）

    可以生成 $\{a^n b^n c^n\}$ 形式的语言
  - 树邻接语法

    TAG： Tree Adjoining Grammar
- 与乔姆斯基语法层级相交叉的语法类别

用什么语法描述自然语言

- 正则语法描述能力太弱、上下文有关语法计算复杂度太高，上下文无关语法使用最为普遍
- 从描述能力上说，上下文无关语法不足以描述自然语言——自然语言中上下文相关的情况非常常见
- 从计算复杂度来说，上下文无关语法的复杂度是多项式的，其复杂度可以忍受
- 为弥补上下文无关语法描述能力的不足 ， 需要加上一些其他手段扩充其描述能力

语法的判定复杂度

- PSG：半可判定

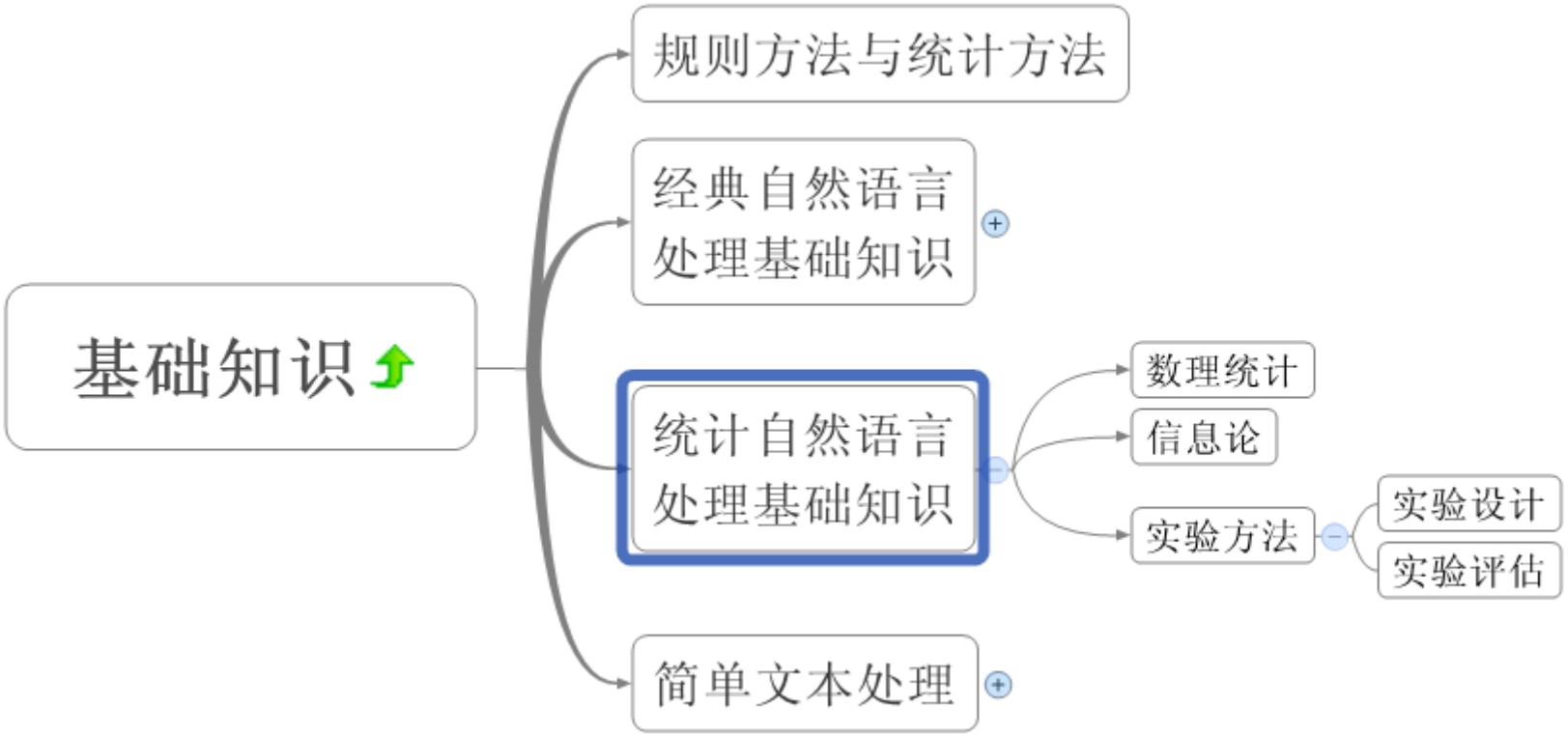
  对于一个属于 0 型语言的句子 L，总可以在确定步内判断出“是”；但对于一个不属于 0 型语言的句子 L’，不存在一个算法，可以在确定步内判断出“否”。

- CSG：可判定，复杂度： NP 完全
- CFG：可判定，复杂度：多项式
- RG：可判定，复杂度：线性

语法、自动机和语言

	语法	自动机	语言	复杂度
0 型	无约束短语结构语法	图灵机	递归可枚举语言	半可判定
		判定器	递归语言	可判定
1 型	上下文有关语法	线性有界自动机	上下文有关语言	NP 完全
	索引	嵌套堆栈自动机	索引语言 适度上下文有关语言	多项式
	树邻接语法	嵌入下推自动机	适度上下文有关语言	多项式
2 型	上下文无关语法	非确定下推自动机	上下文无关语言	多项式
	确定上下文无关语法	确定下推自动机	确定上下文无关语言	线性
3 型	正则语法	有限状态自动机	正则语言	线性

内容提要


基础知识
规则方法与统计方法
经典自然语言处理基础知识
统计自然语言处理基础知识
数理统计
信息论
实验方法
实验设计
实验评估
简单文本处理

概率统计基本概念

- 样本空间
- 事件、随机变量
- 概率
- 条件概率
- 联合概率
- 独立事件
- 贝叶斯原理
- 期望与方差
- 概率分布
- 协方差、相关系数
- 参数估计
- 极大似然估计
- 随机过程
- 马尔科夫过程

信息论基本概念

- 信息量
- 互信息
- 熵
- 联合熵、条件熵
- 交叉熵
- 困惑度、混乱度 Perplexity
- 噪声信道模型

统计自然语言处理方法

- 输入： I
- 输出： O
- 问题： $f(I) \rightarrow O$ （预测）
- 统计模型： $P(O|I)$
- 参数训练： $(I,O) \rightarrow P$
- 问题求解： $O=\max_{O} P(O|I)$

统计模型

- 所谓统计模型，可以理解为一个条件概率分布，就是在给定条件，未知结果的概率分布，可以表示为：

$$P(Result|Condition)$$

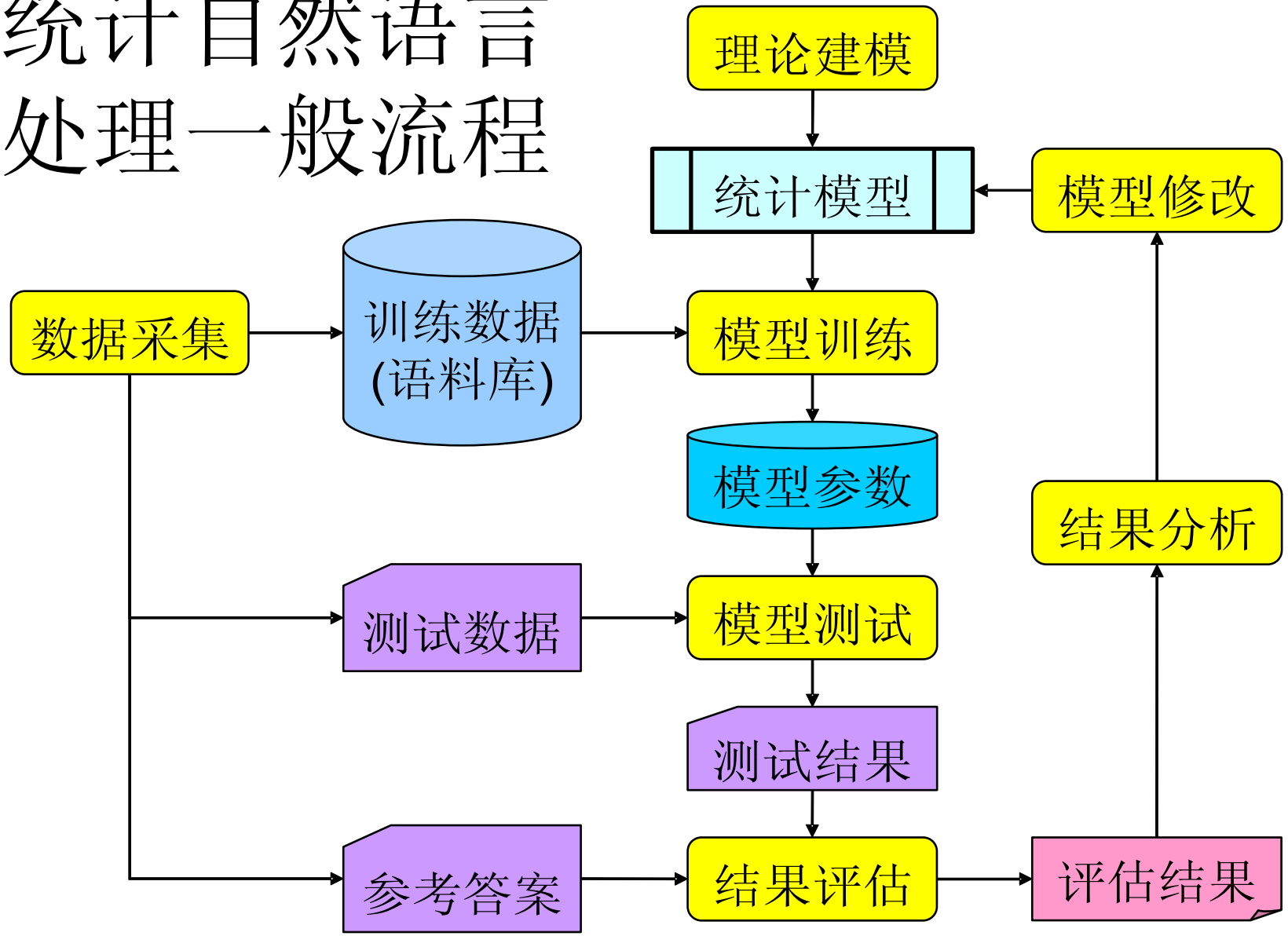
- 有时候我们也采用联合概率形式：

$$P(Result, Condition)$$

- 联合概率可以转换成条件概率：

$$P(Result|Condition)=\frac{P(Result, Condition)}{\sum_{Result'} P(Result', Condition)}$$

统计自然语言处理一般流程



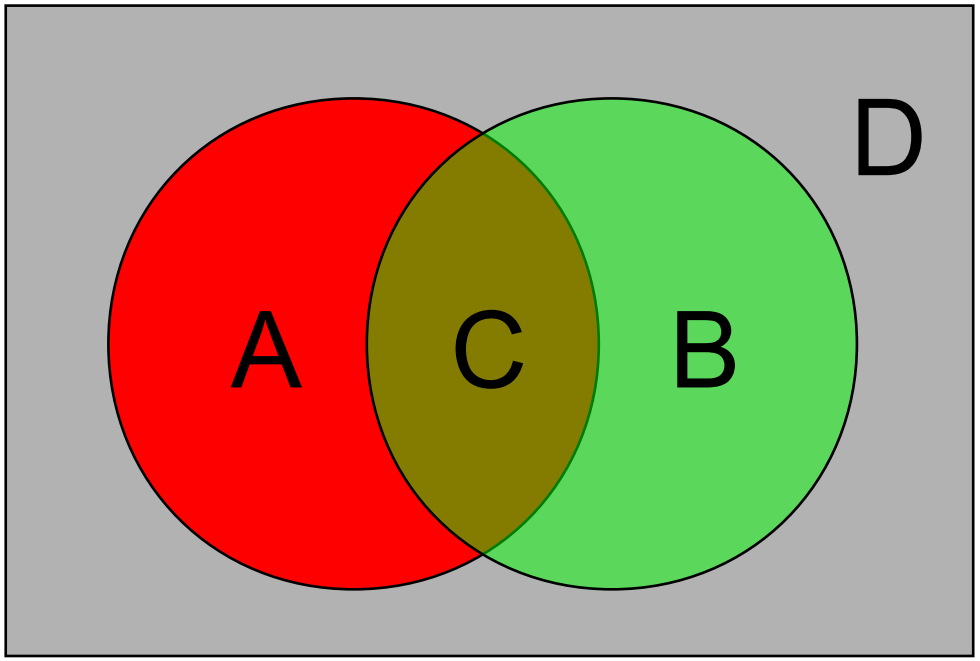
统计模型训练的实验设计

- 训练集 Training Set
  - 用来获得模型参数
- 测试集 Test Set
  - 从训练集以外独立采样
  - 反映系统面对真实世界的处理能力
- 封闭测试与开放测试
- 交叉验证 Cross-Validation
  - 将数据集分为 k 个子集
  - 用 k-1 个子集作训练集， 1 个子集作测试集，然后 k 次交叉验证

性能评价（1）

任务：从所有的测试数据（A+B+C+D）中，
找出满足某种要求的数据（A+C）并加以标注

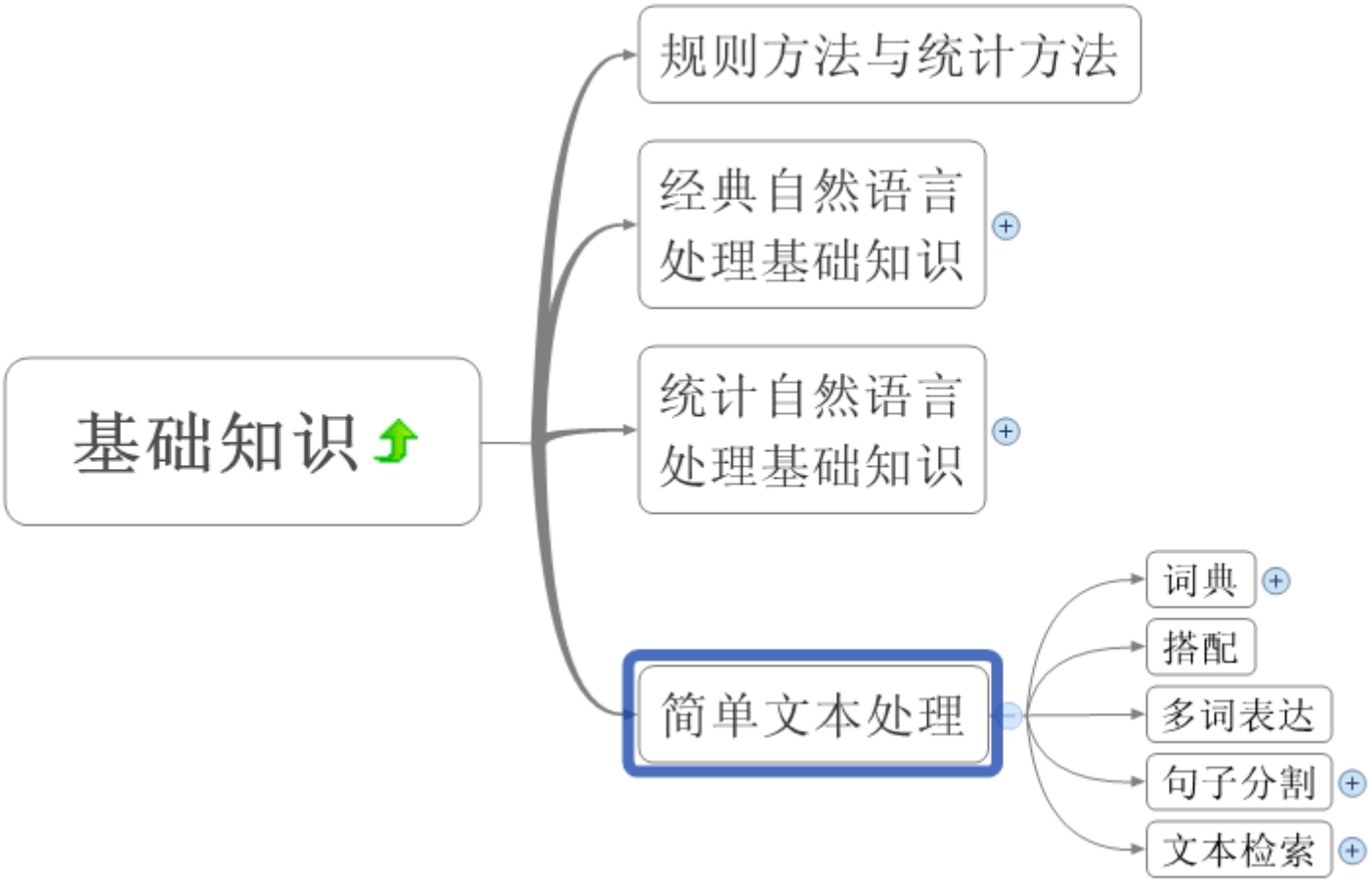
- A＋C：需要标注的数据
- B＋C：系统标注的数据
- C：被正确标注的数据
- A：没有被标注出的数据
- B：被错误标注的数据
- D：不需要标注且系统也没有标注的数据



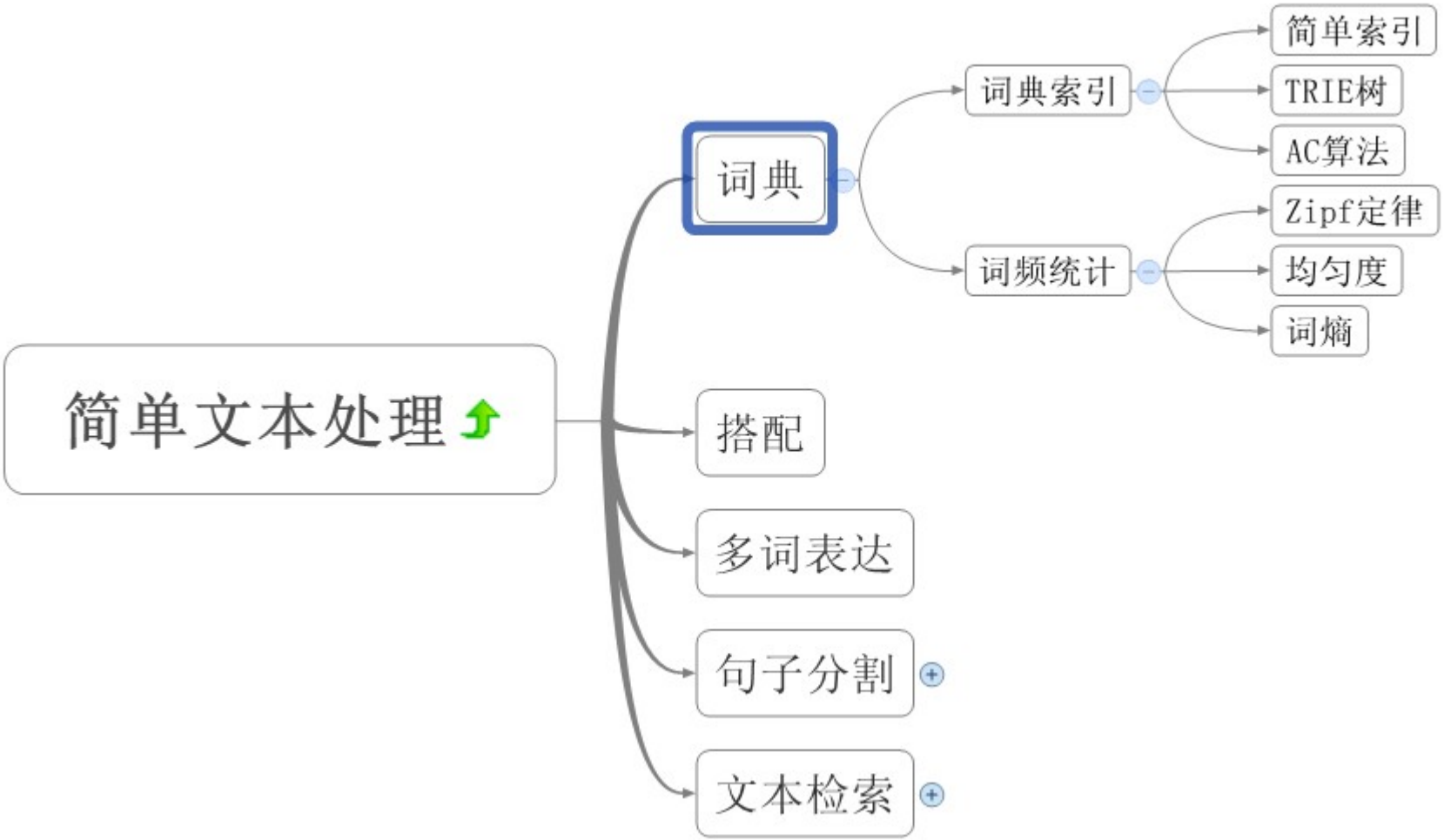
性能评价 (2)

- 准确率 Precision $P=\frac{C}{C+B}$
- 召回率 Recall $R=\frac{C}{C+A}$
- 错误率 Error Rate $E=\frac{A+B}{A+B+C+D}$
- F_β-measure
  通常取 $\beta=1$ $F_\beta=\frac{(\beta^2+1)\times P\times R}{\beta^2\times P+R}$

内容提要


基础知识
规则方法与统计方法
经典自然语言处理基础知识
统计自然语言处理基础知识
简单文本处理
词典
搭配
多词表达
句子分割
文本检索

内容提要


简单文本处理
词典
词典索引
简单索引
TRIE树
AC算法
词频统计
Zipf定律
均匀度
词熵
搭配
多词表达
句子分割
文本检索

索引结构类型

- 整词索引：只支持整词检索
- 分级索引：支持前缀检索
  - 检索句子中某个位置开始的所有词
  - 检索句子中某个位置开始的最长词
- 倒排索引：支持模糊检索

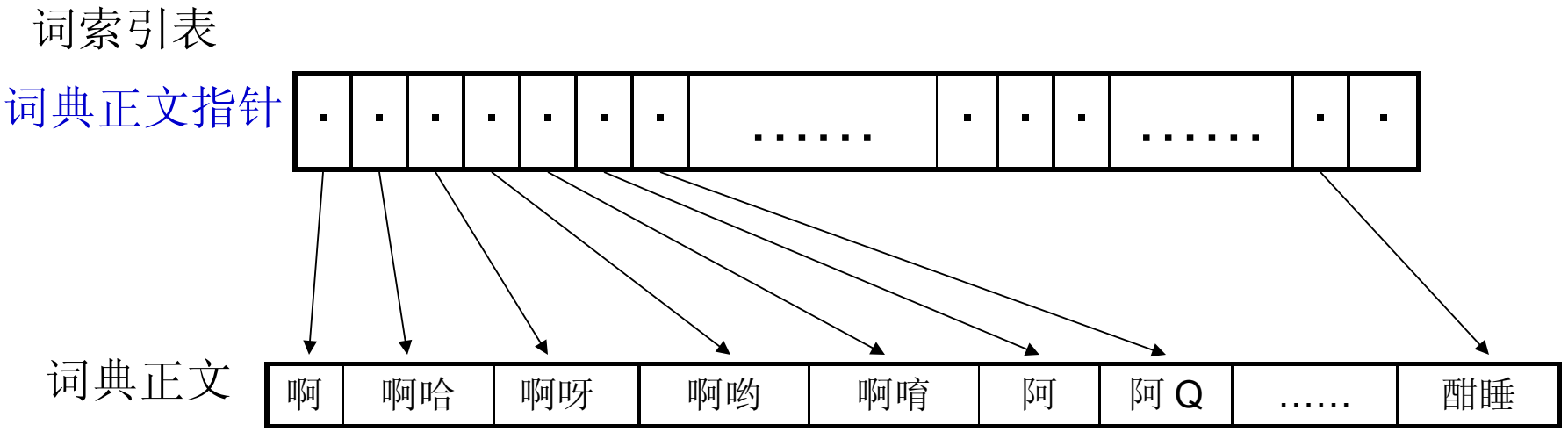
词典检索算法的性能评价

- 时间复杂度
- 空间复杂度
- 增量式索引：词典增加或修改时不用重建全部索引

整词索引

- 只支持整词查找
- 常见索引结构
  - 顺序索引
  - 散列索引
  - 二叉树索引
- 词项的整数化表示

词典顺序索引

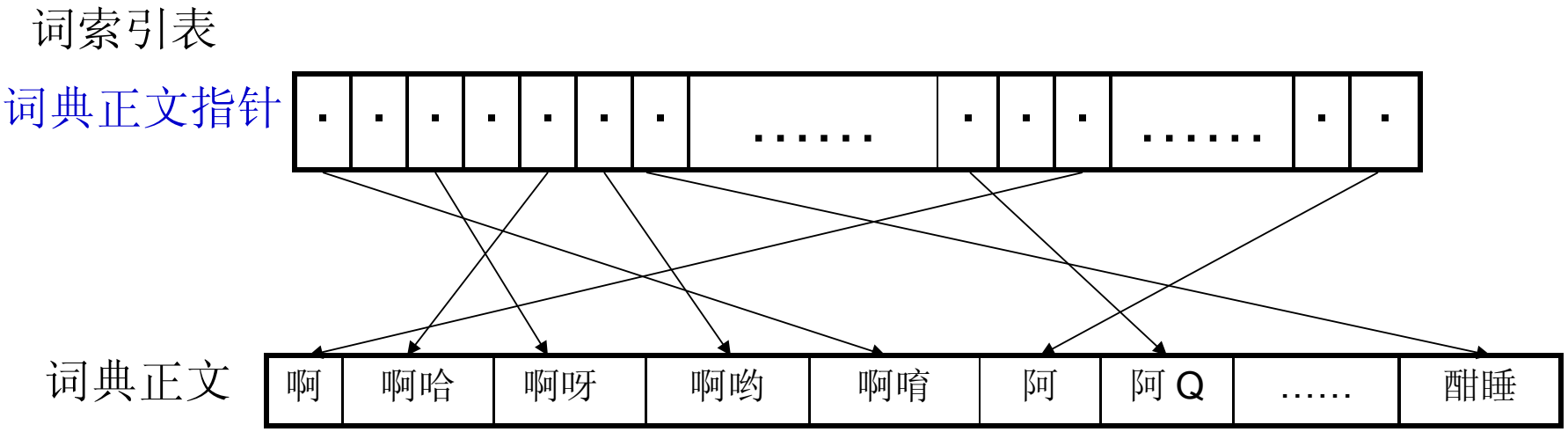


- 索引结构简单，占用空间小
- 不能实现增量式索引：每增加一个词需重新排序

词典顺序索引的查找算法

- 整词二分查找
  - 时间复杂度 $O(\log_2 N)$
  - 无法按前缀查找：查找时精确匹配
- 改进的整词二分查找
  - 时间复杂度 $O(\log_2 N)$
  - 可以实现按前缀查找：查找时匹配前缀，多分枝查找
  - 例子：
    - 查询词：阿拉丁
    - 词典项：阿　阿Q　阿爸　阿拉　阿拉伯　阿拉伯人　阿拉丁　阿拉斯加

词典散列索引



- 索引结构简单，占用空间小（比顺序索引稍大）
- 可以实现增量式索引

词典散列索引的检索算法

- 利用散列（ hash ）函数直接定位
- 效率高：常数
- 不能按前缀查找
- 冲突的解决
  - 使用冲突队列
  - 使用再散列
- 散列函数（ hash ）的选择

二叉树索引

词项的整数化表示

- 词在内存中表示为一个不定长的字符串
- 在很多自然语言处理应用中，为了提高效率，通常将一个词项表示为一个整数，由于整数是定长的，这样可以大大减少存储空间。另外，由于整数比较速度大大快于字符串比较，这样也可以提高系统运行速度。
- 词项整数化需要查表，这种查表可以用整词通过整词所以实现。

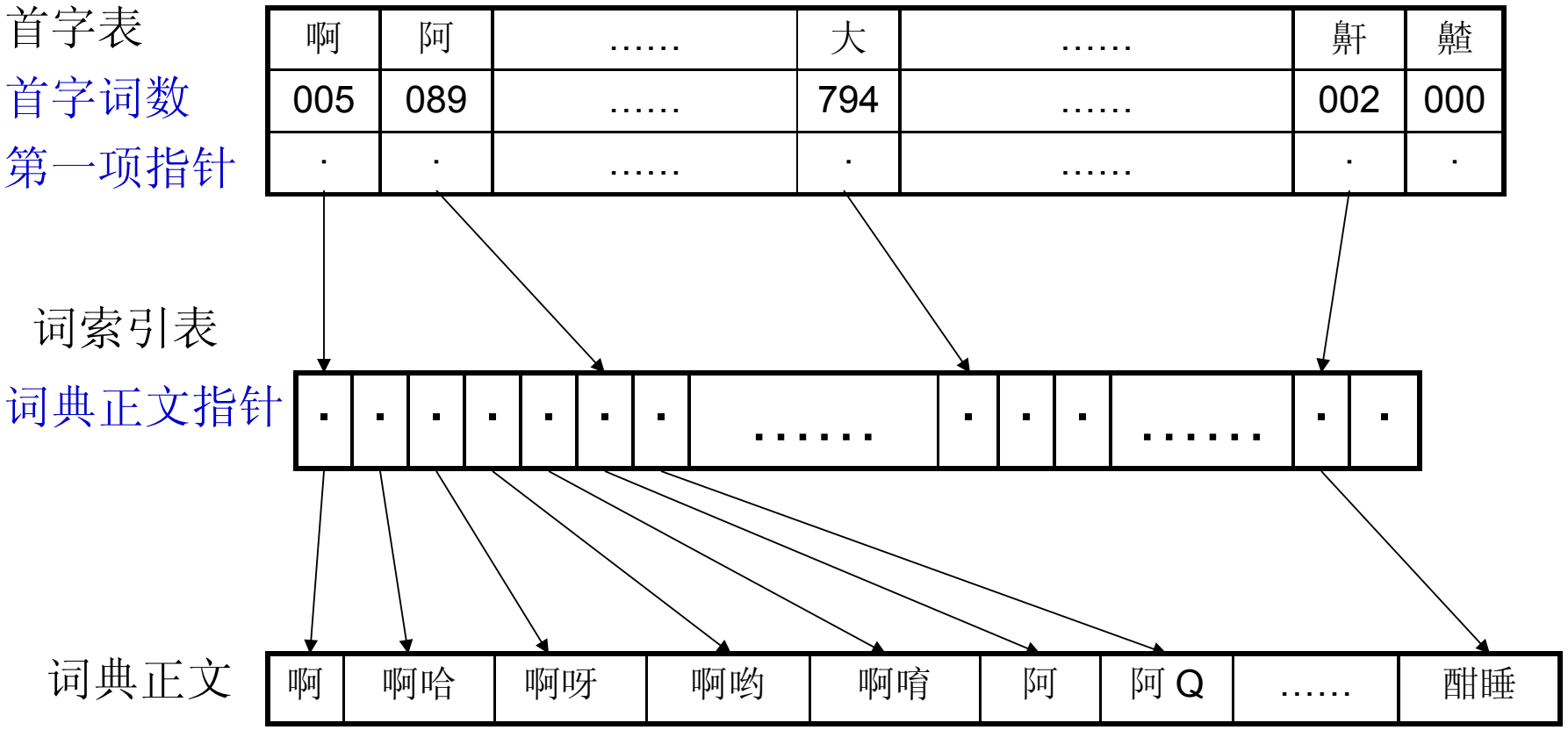
分级索引

- 将词语分成若干部分，为每一部分分别建立索引
- 在分级索引中，每一级索引都可以采用各种不同的索引和查找算法
- 常见分级索引结构：
  - 首字索引
  - TRIE 树
  - 有限状态自动机

首字索引

- 对于汉语而言，第一级索引一般使用词语的首字，所以又常称为首字索引
- 汉语的首字数量有限，可以使用直接定位法，效率最高，空间也不大

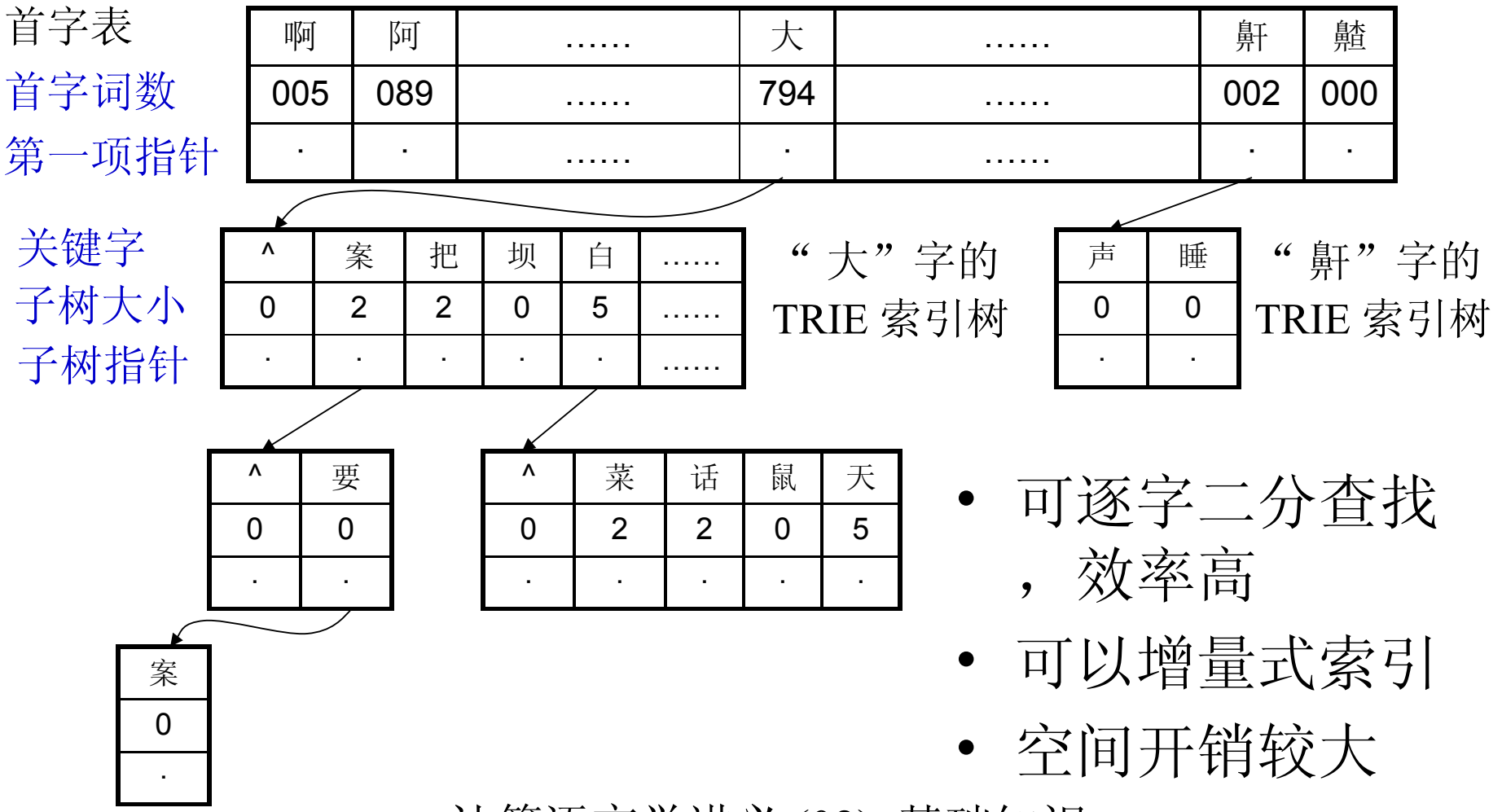
汉语词典按首字顺序索引



首字二分检索

- 时间复杂度： $O(\log_2 N)$
- 空间复杂度： $O(N)$
- 可以按前缀查找
- 不能增量式索引：每次要重新排序

汉语词典 TRIE 树索引



- 可逐字二分查找，效率高
- 可以增量式索引
- 空间开销较大

双数组 TRIE 树

有限状态自动机索引

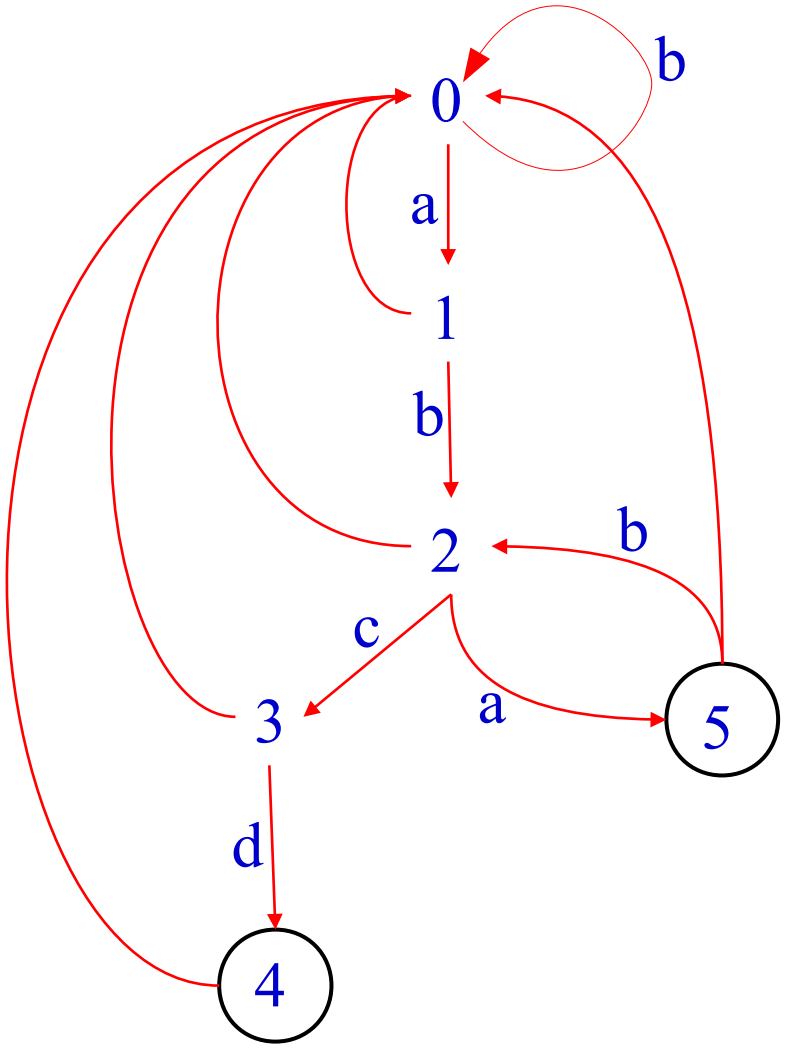
AC 算法 (1)

- 问题
  - 假设词典中有两个词： aba ， abcd
  - 考虑输入串： bababcdab
  - 如何迅速找出输入串中词典词的所有出现？
- 简单解决办法
  - 逐字查词典：效率太低
- AC 算法
  - 将词典构造成一个自动机，一次扫描完成

AC 算法 (2)

aba

abcd

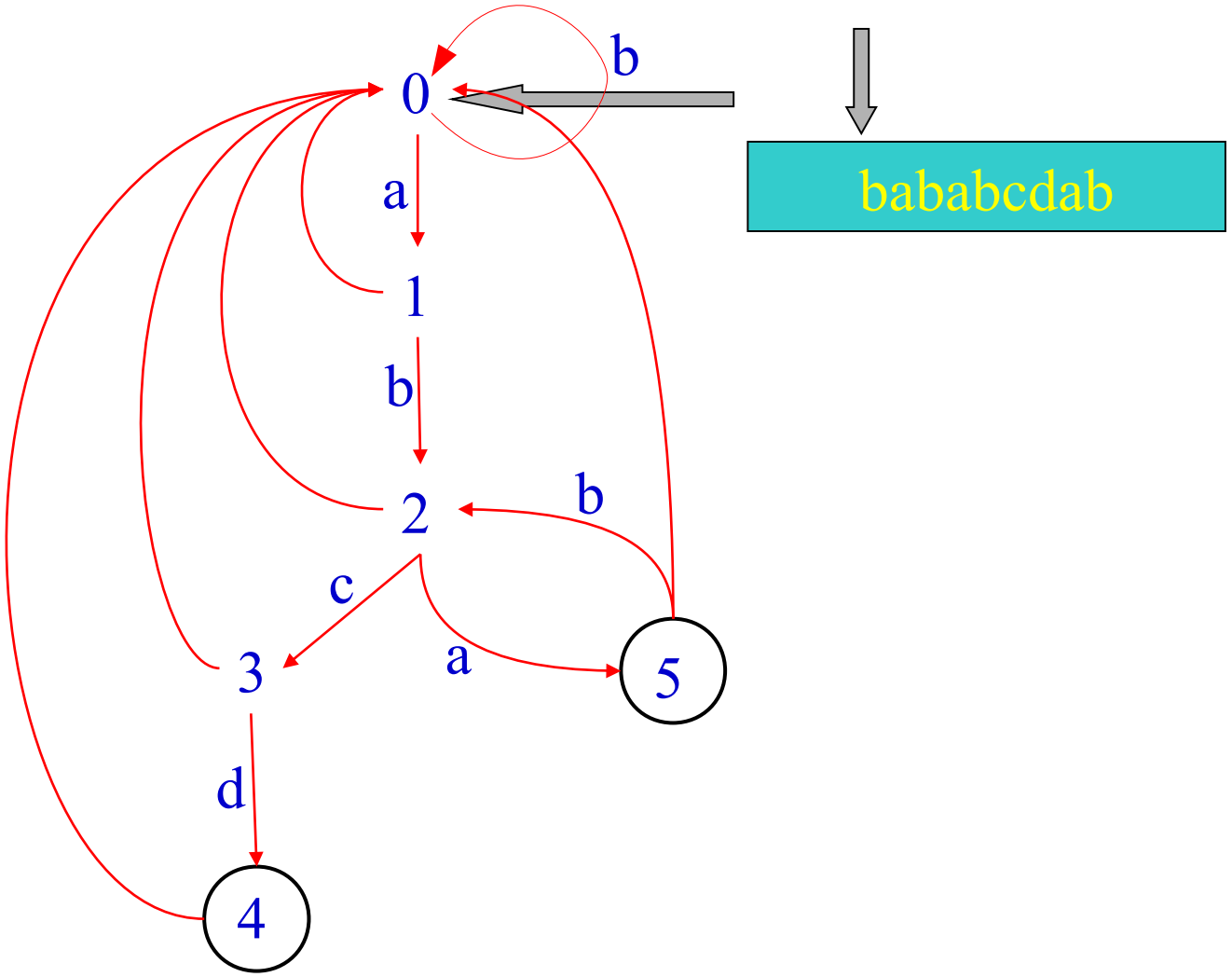


若下一字符没有对应的边，则自动沿空边回到起始状态

AC 算法 (3)

aba

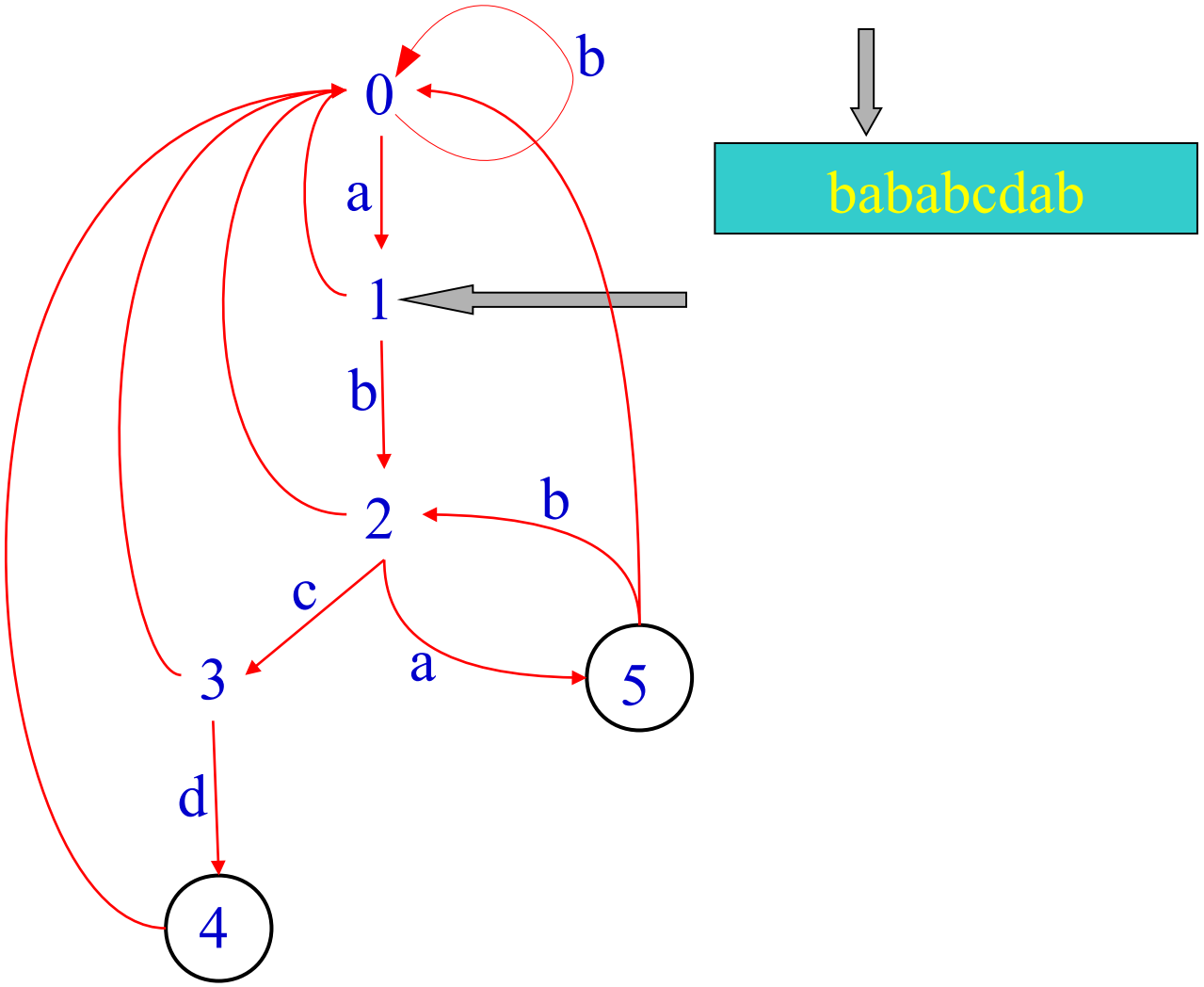
abcd



AC 算法 (4)

aba

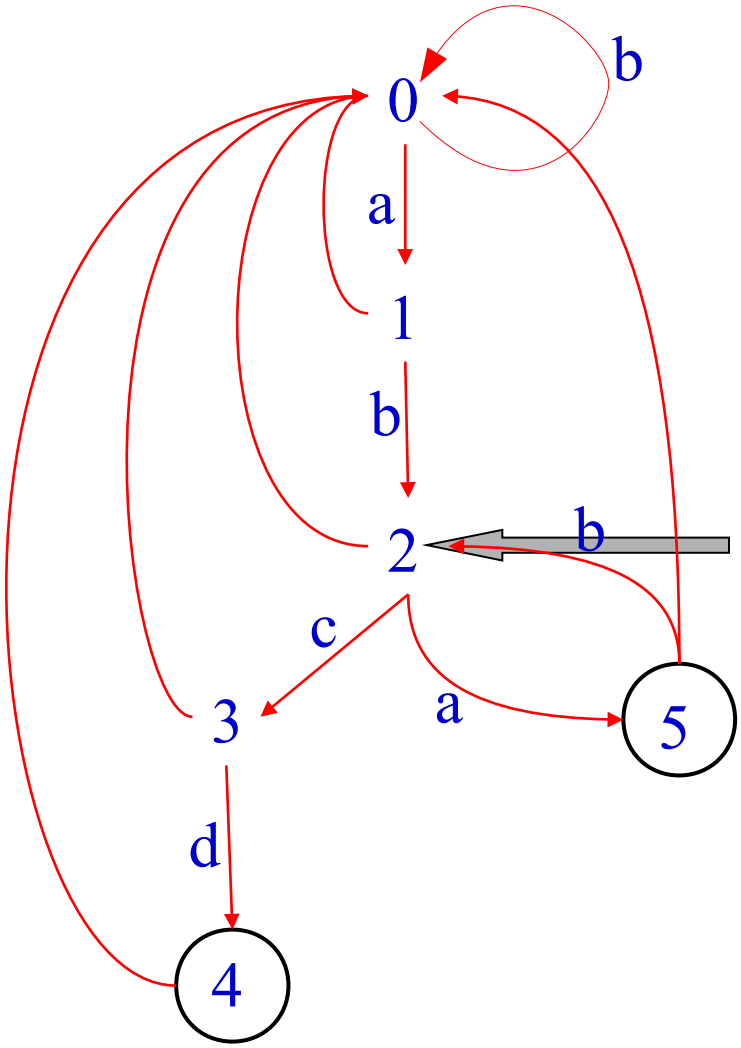
abcd



AC 算法 (5)

aba

abcd

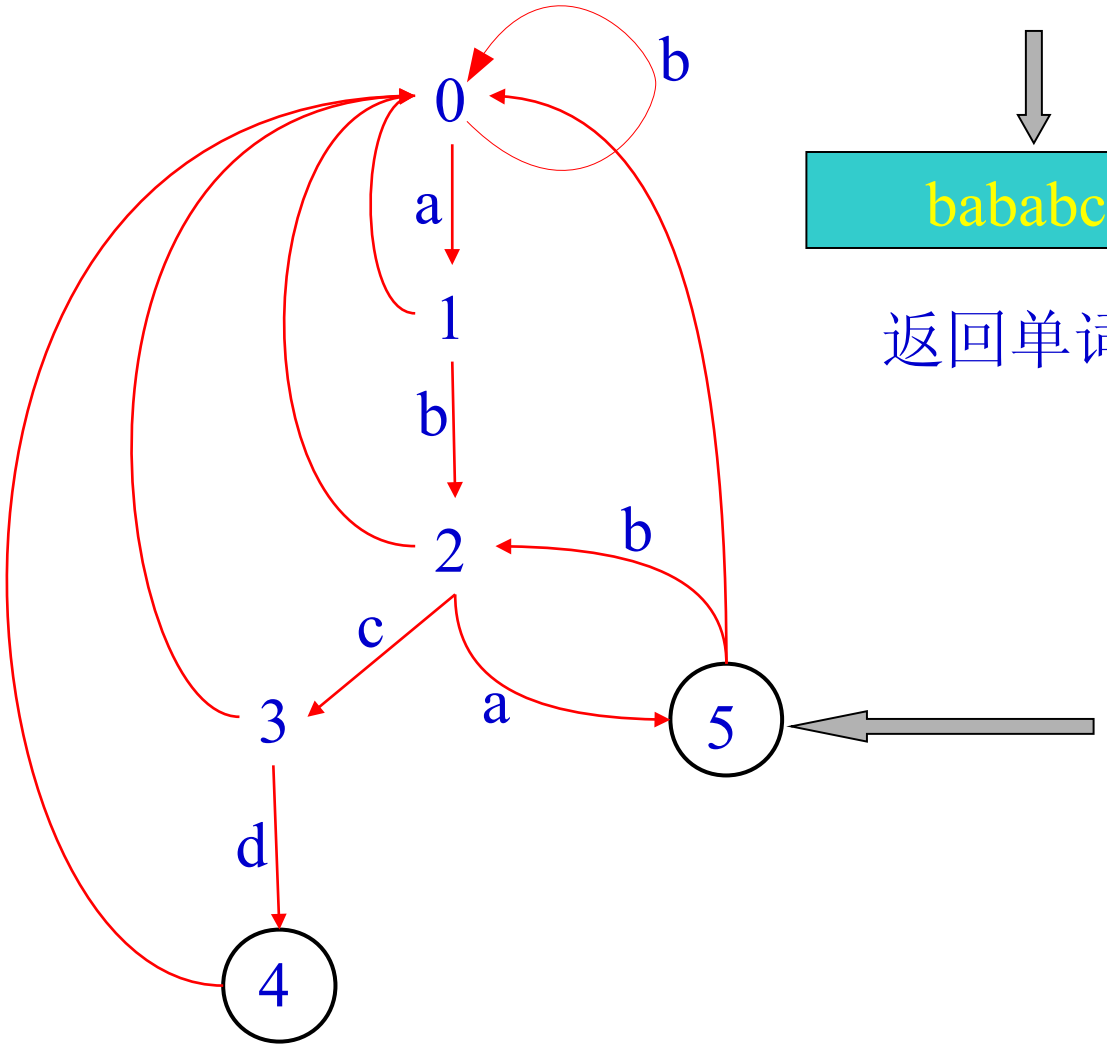


bababcdab

AC 算法 (6)

aba

abcd



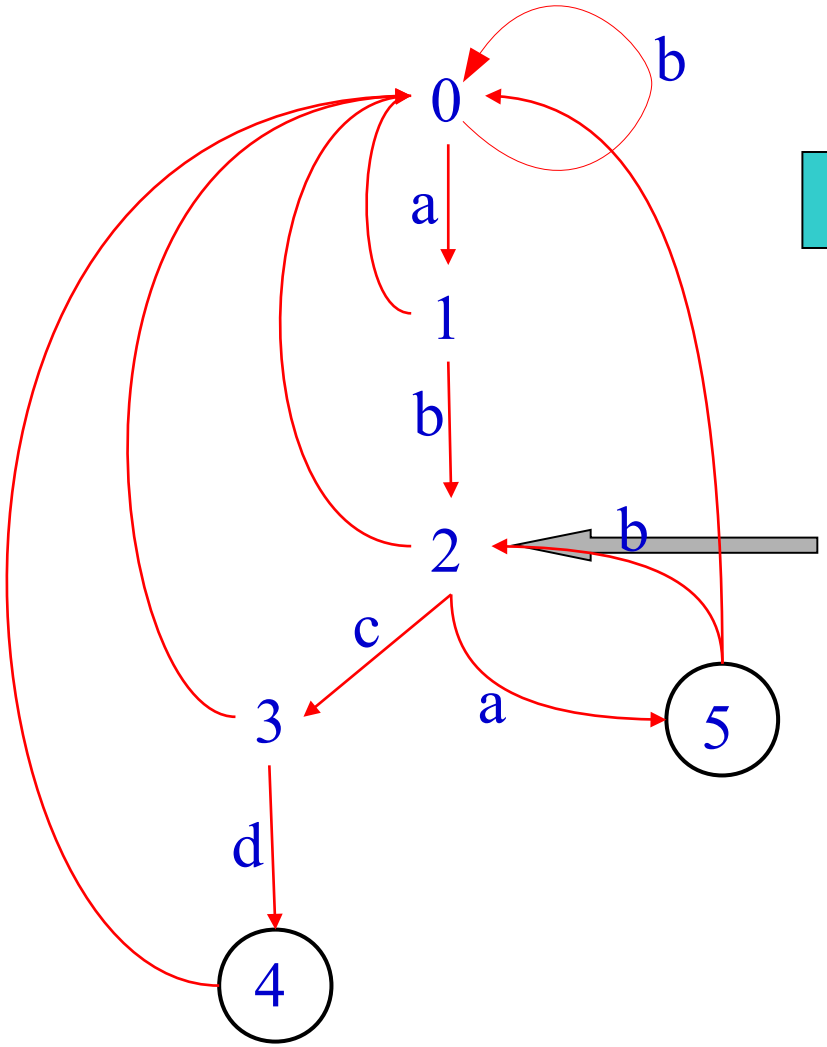
bababcdab

返回单词 aba

AC 算法 （7）

aba

abcd

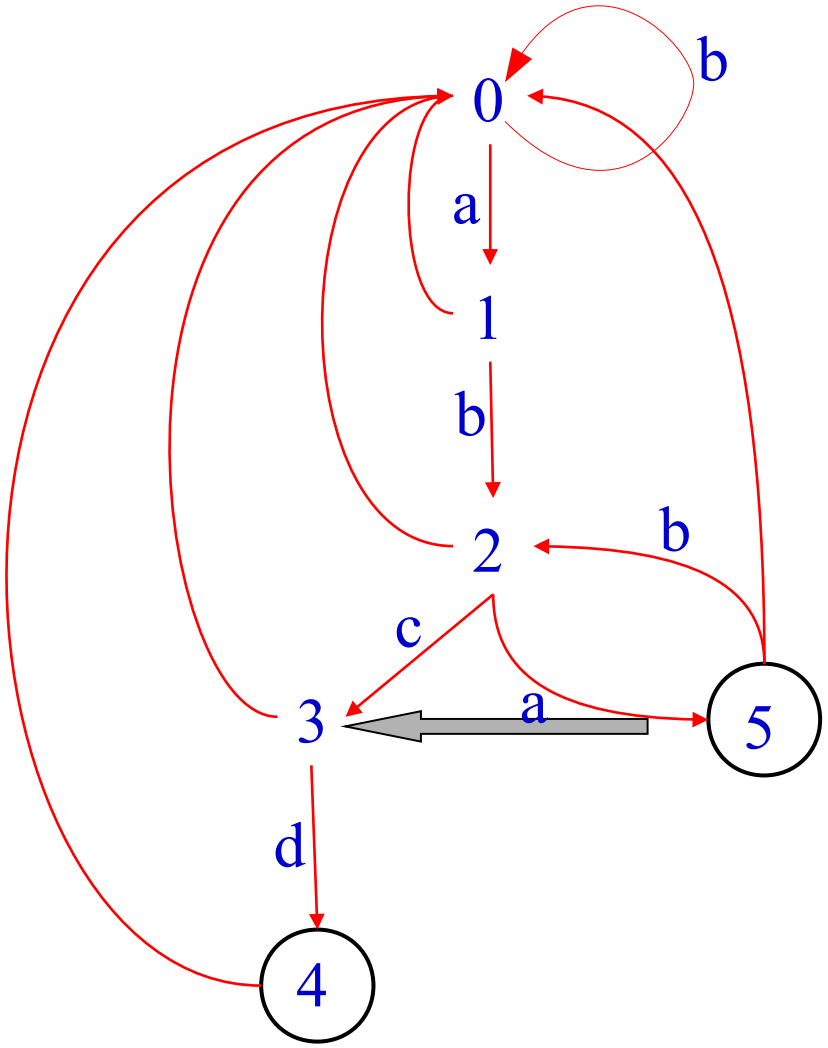


bababcdab

AC 算法 (8)

aba

abcd

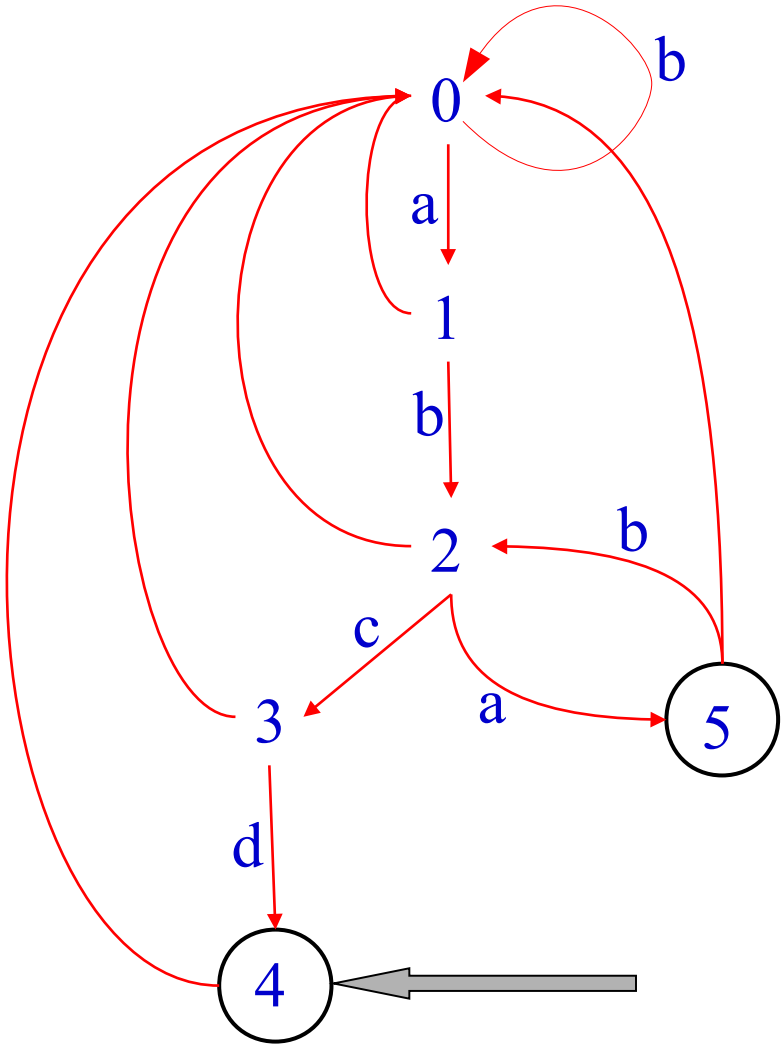


bababcdab

AC 算法 (9)

aba

abcd



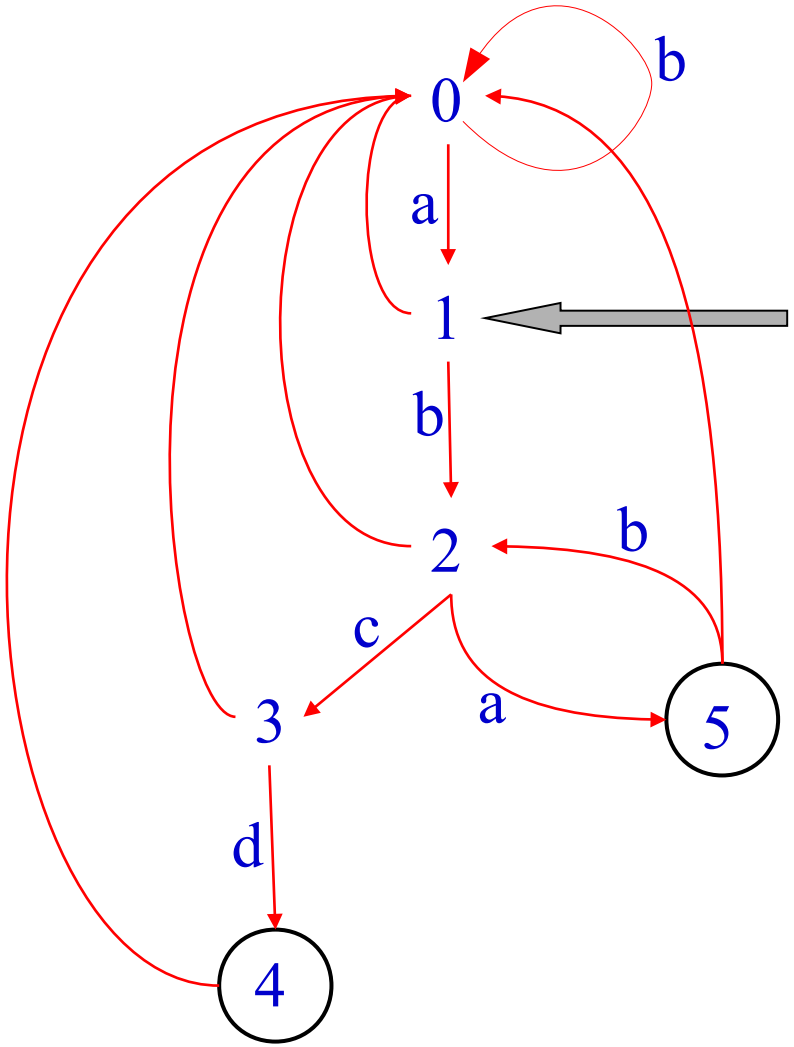
bababcdab

返回单词 abcd

AC 算法 （10）

aba

abcd

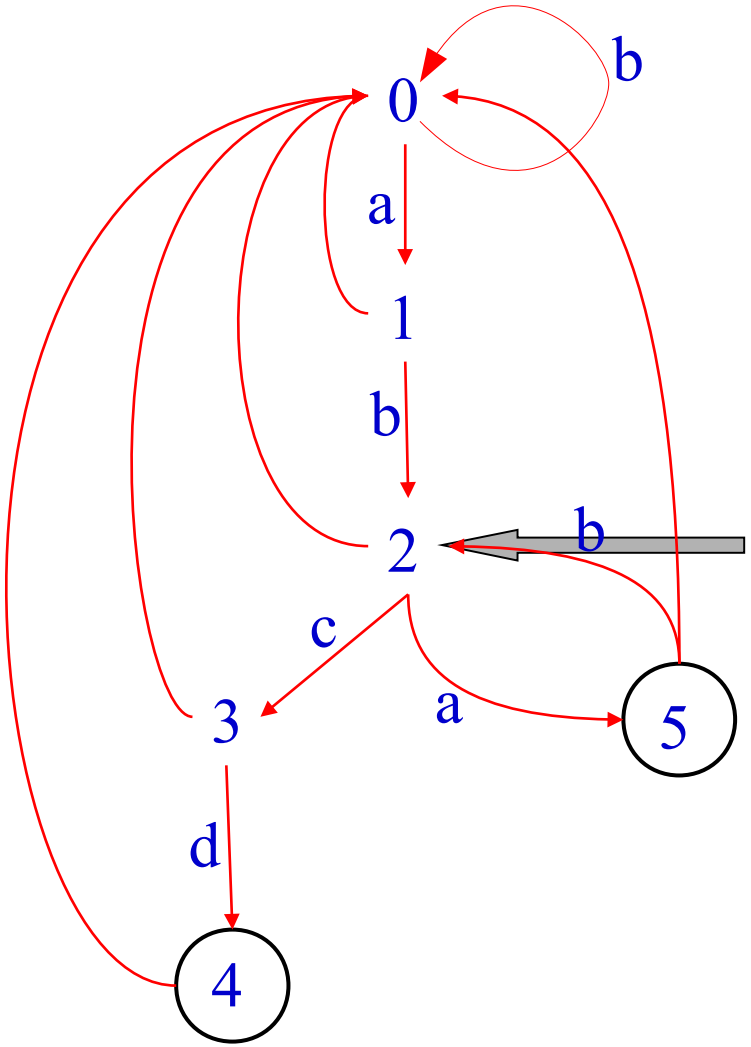


bababcdab

AC 算法 （11）

aba

abcd



bababcdab

AC 算法优缺点

- 优点：速度快，一遍扫描得到所有批评的词语
- 缺点：在词典规模很大时构造自动机的时空开销很大，一般只适合小规模辞典（需确认）

倒排索引

词频统计

- 词频
- Zipf 定律
- 均匀度
- 词熵

词频统计

1998 全年《人民日报》语料中频度最高的 28 个词（带词性）
北京大学计算语言学研究所俞士汶教授提供

word	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	year
的 /u	54454	57239	59625	63097	62699	61038	65511	59153	63778	62656	63729	63823	736802
在 /p	11447	12358	12078	13353	12746	12821	13446	12796	13870	13363	13520	12844	154642
和 /c	10693	10930	12743	11829	12298	12240	13474	12180	12600	13054	12167	13629	147837
了 /u	10235	11291	10682	12143	11654	11659	12625	11449	12357	12535	12799	12355	141784
是 /v	10192	10894	12070	12369	12301	11816	11847	10810	11328	11570	11916	11656	138769
一 /m	7085	7392	7355	8108	7811	7569	8055	7288	7812	7256	7383	6939	90053
不 /d	4536	5107	4928	5700	5510	5483	5426	5000	5444	5163	5427	4967	62691
有 /v	4549	4723	5276	5654	5139	5154	5404	4998	5263	4714	5359	4677	60910
对 /p	3683	4227	4586	4803	4725	4830	4985	4348	4489	4482	4703	4532	54393
中国 /ns	3356	3914	3753	3917	4314	4121	4022	3198	3918	4946	4882	5288	49629
中 /f	3236	3400	3674	3808	3711	3543	4049	3658	3914	3837	3769	3816	44415
这 /r	3205	3505	3617	4030	3792	3673	3747	3570	3815	3519	4357	3289	44119
要 /v	2943	3039	4630	3338	3489	3450	3656	3503	3435	3255	3137	3273	41148
上 /f	3017	3142	3204	3561	3370	3250	3658	3190	3533	3574	3669	3568	40736
他 /r	2826	3793	3034	3584	3555	3036	3303	3112	3751	3845	3382	2420	39641
也 /d	2776	2927	3098	3292	3306	3175	3106	3024	3184	3063	3251	3093	37295
人 /n	2702	2711	2999	3413	3057	2910	3188	3060	3269	2981	3044	2795	36129
个 /q	2633	2507	2865	3213	2912	3007	3237	3046	3015	3213	3164	2957	35769
等 /u	2717	2642	2746	2966	2963	2876	3378	2883	2966	3053	3097	2935	35222
说 /v	2583	2845	3466	2911	3096	2975	2851	2549	2833	2874	2930	2441	34354
经济 /n	2597	2431	3094	2718	2415	2603	2601	2602	2685	3106	3497	3480	33829
就 /d	2282	2633	2522	2996	2681	2643	2763	2753	2805	2739	2806	2320	31943
工作 /vn	2120	2458	3474	2267	2540	2685	2680	2786	2822	2533	2588	2849	31802
年 /q	2416	2365	2890	2699	2353	2407	2661	2122	2577	2799	2900	3444	31633

齐普夫（Zipf）定律

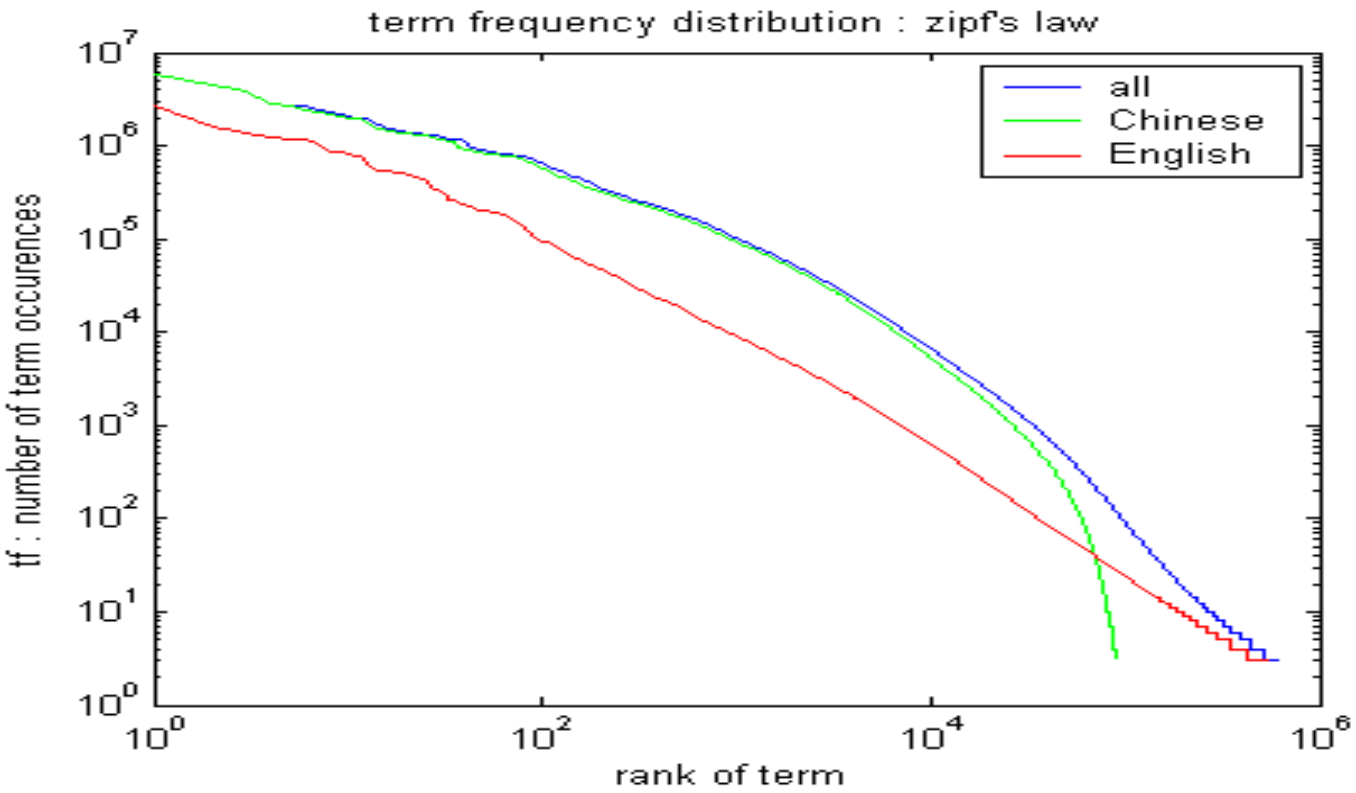
- 齐普夫定律是美国学者G.K. 齐普夫于本世纪40年代提出的词频分布定律。它可以表述为：

  如果把一篇较长文章中每个词出现的频次统计起来，按照高频词在前、低频词在后的递减顺序排列，并用自然数给这些词编上等级序号，即频次最高的词等级为1，频次次之的等级为2，……，频次最小的词等级为D。

  若用f表示频次，r表示等级序号，则有：

$$f \propto \frac{1}{r}$$

齐普夫（Zipf）定律—实验



汉语和英语的词频分布曲线
根据大规模语料库进行自动分词产生

齐普夫（Zipf）定律

- 根据图形分析，发现以下问题：
  - 英语词频分布曲线中段比较符合 Zipf 定律，但高频段和低频段略有降低
  - 汉语低频段曲线急遽下降
- 分析原因
  - Zipf 曲线对高频词和低频词刻画不够准确
  - 对于汉语来说，导致曲线变形的主要原因是汉语分词软件的新词语识别功能不满足要求，大量低频新词语无法识别，使得词汇总量受到限制

Mandelbrot 定律

- 对 Zipf 定律的修正：

$$f = P(r + \rho)^{-B}$$

- Mandelbrot 对词频分布规律的描述更为准确，特别是对高频词和低频词

齐普夫（Zipf）定律—其他

- 一个词的语义数目服从以下法则：

$$m \propto \sqrt{f}$$

- 一个词的词频和词的长度成反比
- ……

词语分布均匀度

1998 年《人民日报》语料库部分词语分布
北京大学计算语言学研究所俞士汶教授提供

Word	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.	Year
编织袋 /n	2	2	3			6	12	109	23		5	2	164
舰长 /n	3	9					75	36			1	2	126
九江 /ns	1	4	11	6	5	12	60	230	83	52	11	4	479
九江市 /ns	2	3	8	1	2	4	16	59	34	5	4	0	138
抗洪 /v		2	3	4	5	10	145	1125	620	317	44	45	2320
抗洪 /vn	1	5	1	1	7	10	211	2004	1326	852	140	66	4624
抢险 /v	20	8	15	10	8	11	98	807	411	220	12	32	1652
抢险 /vn	10	5	4	17	9	11	105	893	497	281	22	19	1873
水情 /n				1		3	16	61	28	14	2	3	128
滔滔 /z	3	4	1	3	2	1	16	65	46	27	6	5	179
新加坡 /ns	42	80	38	37	41	40	60	57	41	71	89	31	627

词语分布均匀度：如何衡量？

- 熵？
- 北大张化瑞博士提出匀度公式：

$$res = \left(\frac{\sqrt{F_1} + \sqrt{F_2} + ... + \sqrt{F_n}}{n} \right)^2$$

$$E = \frac{F_1 + F_2 + ... + F_n}{n}$$

$$DC = \frac{res}{E}$$

词语均匀度

1998 全年《人民日报》语料中分布均匀度最高的 28 个词
北京大学计算语言学研究所俞士汶教授提供

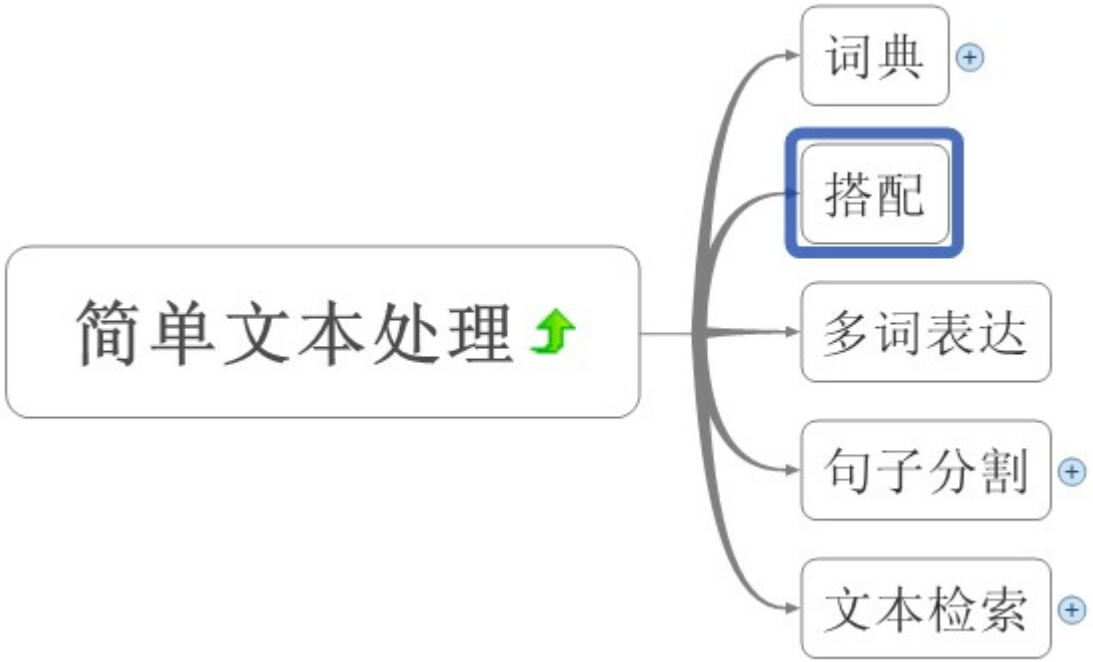
Word	POS	Frequency	DC
中	f	44418	0.99989
的	u	736812	0.99988
在	p	154681	0.99985
为	p	31394	0.99971
等	u	35223	0.99970
上	f	40645	0.99968
个	q	35769	0.99968
了	u	141789	0.99968
能	v	19574	0.99966
也	d	37295	0.99964
从	p	26576	0.99960
大	a	27858	0.99952
和	c	147835	0.99951
并	c	18585	0.99950

Word	POS	Frequency	DC
都	d	24309	0.99949
人	n	36128	0.99949
对	p	54411	0.99946
不	d	62676	0.99942
把	p	19989	0.99942
是	v	138767	0.99942
到	v	30611	0.99940
记者	n	28324	0.99936
地	u	28010	0.99935
有	v	60901	0.99935
一	m	90038	0.99934
结束	v	3255	0.99931
通过	p	8174	0.99928
那些	r	1871	0.99928

字母熵、汉字熵、词熵

- 词熵
  - 词熵是针对一种语言而言的
  - 一种语言的所有词语的概率分布构成的熵称为该语言的词熵
  - 冯志伟计算出：
    - 汉字的熵为 9.65 比特
    - 法语一个字母的熵为 3.98 比特
    - 意大利语一个字母的熵为 4.00 比特
    - 西班牙语一个字母的熵为 4.01 比特
    - 英语一个字母的熵为 4.03 比特
    - 德语一个字母的熵为 4.12 比特
    - 俄语一个字母的熵为 4.35 比特

内容提要



（修改提示）

- 搭配和重复串识别这两部分内容可以加强
- 搭配可以泛化成相关度
- 可以考虑引入相似度内容
- 整个第二讲内容可以扩充，第三讲内容适当压缩

搭配

- 什么是词语搭配
- 搭配的类型
- 搭配抽取
- 搭配的统计度量方法

什么是词语搭配

- 如果两个或者两个以上的词语经常共现，并且表达特定的语义含义，那么我们称这些词语构成一个搭配

词语搭配的强度

- 固定搭配：不可替换，形式固定
  - 一石两鸟，一箭双雕
  - 缘木求鱼，釜底抽薪
  - 刻舟求剑 but not 刻船求剑
- 强搭配：不可替换，形式可变
  - 匍匐前进
- 弱搭配：一定程度可替换
  - 减少职位，缩减职位
  - 油价上升，股市上升

词语搭配抽取

- 基于接续的搭配抽取
  - 统计经常出现的连续词串
- 基于窗口的搭配抽取
  - 以一个词为中心，统计左右一定范围的词语窗口内出现频度较高的词语
- 基于句法的搭配抽取
  - 根据句法结构，统计在同一句法关系中频繁出现的词语组合

词语搭配的统计度量

- 相对词频
- DICE 系数
- 互信息（ MI， Mutual Information ）
- T 测试（ T-Test ）（暂缺）
- 卡方（ Chi-Square ）测试
- 对数似然比（ LLR， Log-Likelihood Ratio ）

相对频率

$$p(w_1|w_2)=\frac{freq(w_1,w_2)}{freq(w_2)}$$

$$p(w_2|w_1)=\frac{freq(w_1,w_2)}{freq(w_1)}$$

DICE 系数

$$DICE(w_1|w_2)=\frac{2\,freq(w_{1,}w_2)}{freq(w_1)+freq(w_2)}$$

互信息

- 互信息（mutual information）
  通过两个事件X和Y各自出现的概率为p(X)和p(Y)，他们联合出现的概率为p(X,Y)，这两个事件之间共同的互信息量定义为：

$$I(X,Y)=-\log_2\frac{p(X)p(Y)}{p(X,Y)}$$

- 当两个事件相互独立时，互信息量为0；
- 当两个事件倾向于同时出现时，互信息量为正；
- 当两个事件倾向于互相排斥时，互信息量为负；
- 利用互信息作词语相似度计算效果较差。

T 测试

卡方测试

χ^2 (chi-square)

利用联立表（ contingency table ）

	Wt+	Wt-
Ws+	31,950(a)	12,004(b)
Ws-	4,793(c)	848,330(d)

$$\chi^2 = \frac{(ad-bc)^2}{(a+b)(a+c)(b+d)(c+d)}$$

对数似然比

- 对数似然比（Log Likelihood Ratio, LLR）

$$LLR = \log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) - \log L(p, k_1, n_1) - \log L(p, k_2, n_2)$$

其中：$\log L(p, n, k) = k \log p + (n-k) \log(1-p)$

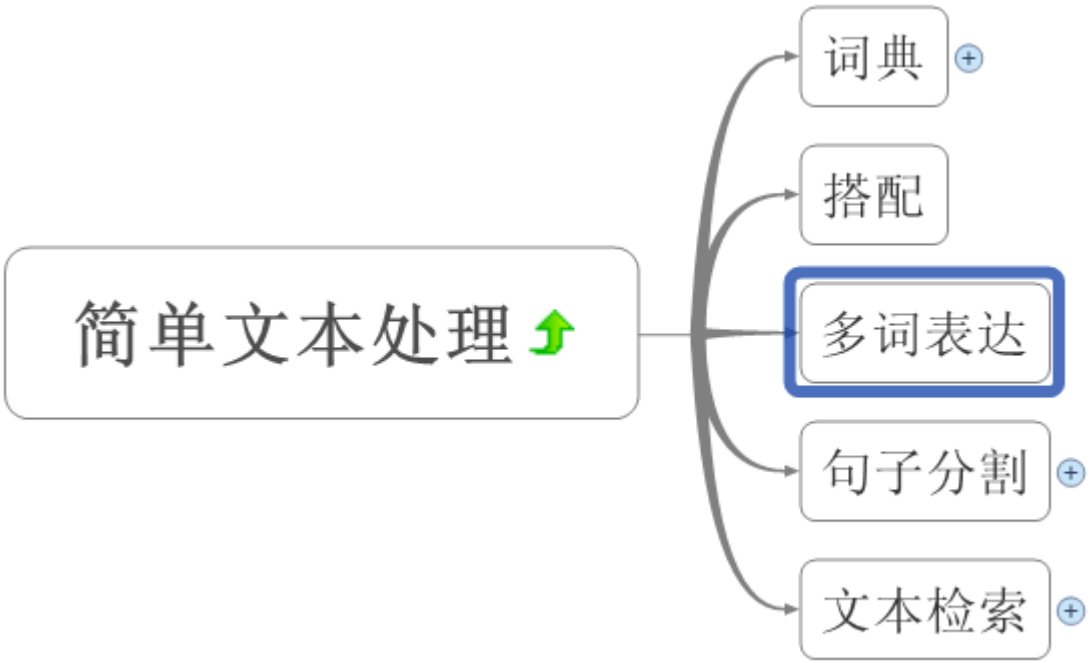
$$k_1 = f(w_t, w_s), k_2 = f(w_t, \neg w_s), n_1 = f(w_s), n_2 = f(\neg w_s)$$

$$p_1 = p(w_t | w_s) = \frac{k_1}{n_1}, p_2 = p(w_t | \neg w_s) = \frac{k_2}{n_2}, p = p(w_t) = \frac{k_1 + k_2}{n_1 + n_2}$$

词语搭配度量

- 相对词频：会偏向于高频词
- 互信息：对于稀疏数据存在过高评价
- 卡方测试：数据稀疏情况下表现不好
- 对数似然比：可以识别稀疏二元搭配

内容提要



多词表达

- 多词表达（MWE： Multi-Word Expression）
- 重复子串
- 有意义串
- 术语

重复子串检测

- Makoto Nagao 的算法
  - Makoto Nagao, Shinsuke Mori. A new method of N-gram statistics for large number of n and automatic extraction of words and phrases from large text data of Japanese. Proceedings of ACL-1994[J], 1994
  - 吕学强对 Nagao 算法的改进：吕学强，面向机器翻译的 E-Chunk 获取与应用研究，第三章 单语 Chunk 获取方法，东北大学博士论文， 2003 年
  - 复杂度： O(n*log(n))， n 是文本长度，算法限制较多
- 后缀树算法
  - 后缀树算法比较成熟，在很多领域都有应用
  - 后缀树构造时间与字符表大小有关，字符表较小时最小可到 O(n)，但对于大字符表的情况时间复杂度有所提高
- 邹纲的方法
  - 邹纲，中文新词语自动检测研究，第四章 重复串查找，中国科学院计算技术研究所硕士论文， 2004
  - 复杂度： O(k*n)， n 是文本长度， k 是最长重复串长度，算法限制较少

有意义串判别

- 反例：词典过滤
- 规则：过滤规则
- 内部统计信息：
  - 互信息 (MI)
  - 卡方 (Chi-Square)
  - 对数似然比 (LLR)
- 外部统计信息：上下文熵
- C-Value

上下文熵

$$LE(W) = -\sum_{x_i \in X} \frac{C(x_i, W)}{N(W)} \log \frac{C(x_i, W)}{N(W)}$$

$$RE(W) = -\sum_{y_i \in Y} \frac{C(W, y_i)}{N(W)} \log \frac{C(W, y_i)}{N(W)}$$

C-Value

$$C\text{-}value(p) = \begin{cases} (L(p)-1) * F(p) & \text{当} N(p) = 0\text{时} \\ (L(p)-1) * (F(p) - S(p)/N(p)) & \text{其他} \end{cases}$$

$L(p)$：词串p的长度；

$F(p)$：词串p在语料库中出现的频率；

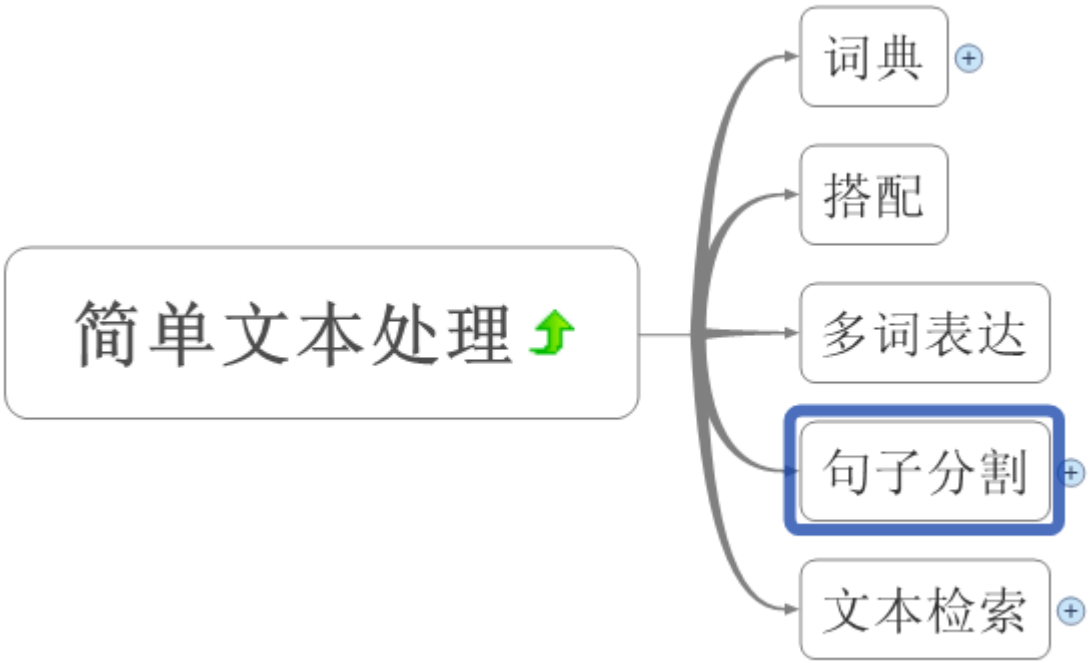
$S(p)$：词串p出现在其他候选长词串中的频率；

$N(p)$：包含词串p的候选长词串的数目。

不仅考虑词串本身频率，而且考虑词串作为其他词串的字串的频率

在其他词串中经常出现的词串更重要

内容提要



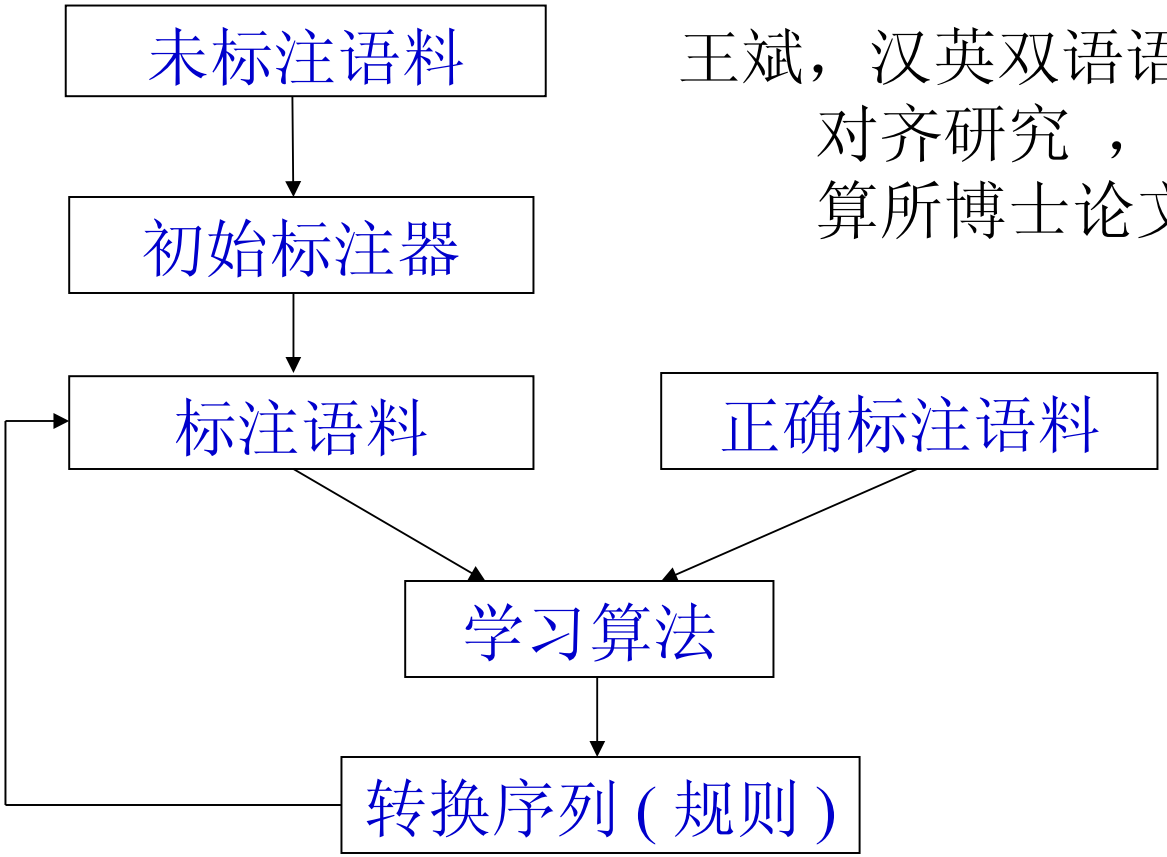
英语句子边界的识别

- 困难所在：句点的多义性
- 例子：
  - The group includes Dr. J.M. Freeman and T. Boone Pickens Jr.
  - “This issue crosses partly lines and crosses philosophical lines!” said Rep. John Rowland (R., Conn.).
  - The machine is 1.1 kilograms weight.
  - Our web site address is http://www.ict.ac.cn .
  - One of them looked sideways with his drunken eyes and said, ″You...you are Minister Feng?″

英语句子边界的识别

- 分类问题
- 解决办法：各种机器学习算法均可采用
  - 转换学习方法

英语句子边界的识别
基于转换的错误驱动的学习 （1）



王斌，汉英双语语料库自动对齐研究 ，中科院计算所博士论文， 2000

英语句子边界的识别
基于转换的错误驱动的学习 (2)

- 激发环境

  〖左单词〗〖前缀〗〖句点〗〖后缀〗〖右单词〗

  leftword prefix .
  suffix rightword

- 例子

  I have never seen Mr. Wang since 1994.

  “Mr” 后的句点的以上各值为：

  leftword=“seen”, prefix=“Mr”,

  suffix=NULL, rightword=“Wang”

英语句子边界的识别
基于转换的错误驱动的学习 (3)

- 激发环境函数（布尔值）
  - prefix：是否为空(isNull)、右端是否数字(isRdigit)、右端是否句点(isRdot)、右端是否其他标点(isRpunct)、是否英语单词(isEnglishword)、首字母是否大写(isCapitalized)
  - suffix: 是否为空、左端是否数字(isLdigit)、左端是否句点(isLdot)、左端是否其他标点(isLpunct)、是否英语单词、首字母是否大写
  - leftword：是否为空、是否英语单词、左端是否句点、左端是否其他标点符号、首字母是否大写
  - rightword: 是否为空、是否英语单词、右端是否句点、右端是否其他标点符号、首字母是否大写

英语句子边界的识别
基于转换的错误驱动的学习 (4)

- 转换规则模板

  如果：某句点的 prefix 各属性取值为布尔集合 A,

  同时 suffix 各属性取值为布尔集合 B，
  leftword 各属性取值为布尔集合 C, rightword
  各属性取值为布尔集合 D

  那么：该句点将由表示结尾改变为不表示结尾
  （或者由不表示结尾改为表示结尾）

  其中 A 、 B 、 C 、 D 分别表示各属性的取值集合

英语句子边界的识别
基于转换的错误驱动的学习 (5)

- 初始标注器
  - 任意标注，或者
  - 全部句点标注为句子边界，或者
  - 全部句点标注为非句子边界
- 规则组织形式：一个规则序列
- 标注算法：对初始标注的语料中的每一个句点，依次执行规则序列中的每一条规则，对该句点的标注进行修改

英语句子边界的识别
基于转换的错误驱动的学习 （6）

- 规则评价函数

$$F(T) = \begin{cases} Num\_good\_transform_T, \\ \qquad 当 Num\_bad\_transform_T = 0 \\ Num\_good\_transform_T / Num\_bad\_transform_T, \\ \qquad 当 Num\_bad\_transform_T \neq 0 \end{cases}$$

英语句子边界的识别
基于转换的错误驱动的学习 (7)

- 学习算法的基本思想：
  - 每次循环时，在所有可能应用的规则中，寻找一条最好的规则，使其评价函数最高
- 开放测试结果（2.2M 训练语料）：

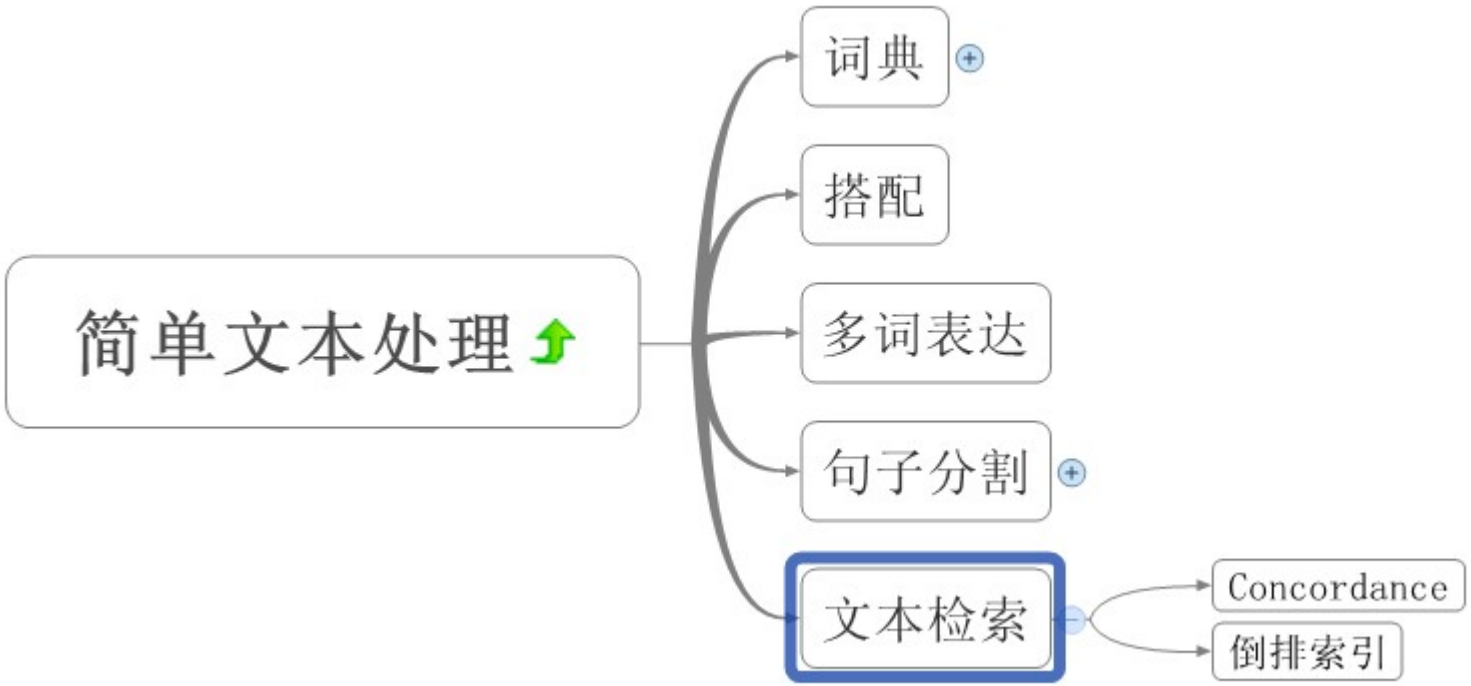
	测试集 1	测试集 2	测试集 3
文章类型	计算机文献	英文小说	杂类
句点个数	1757	814	4060
正确率	98.1%	97.2%	97.9%

英语句子边界的识别
基于转换的错误驱动的学习 (8)

- 学习到的规则样例

if		
	prefix 满足	isRdigit=Yes
	suffix 满足	isLdigit=Yes
	leftword 满足	isEnglishword=Yes, isRdot=No, isRpunc=No, isCapitalized=No
	rightword 满足	isEnglishword=Yes, isLdot=No, isLpunc=No, isCapitalized=No
then		
	该句点表示结尾→该句点不表示结尾	

内容提要



文本检索

- Concordance （应补充）
- 倒排索引