

# Web信息处理与应用



## 第三节 网页文字处理

徐童 2023.9.18

- 重复文档检测与N-gram模型
- 近似重复文档的检测方法：指纹表示法
  - 1) 对文档进行分词处理，并进行n-gram组合
  - 2) 挑选部分n-gram用于表示这一文档
  - 3) 对被选中的n-gram进行散列
  - 4) 存储散列值作为文档指纹

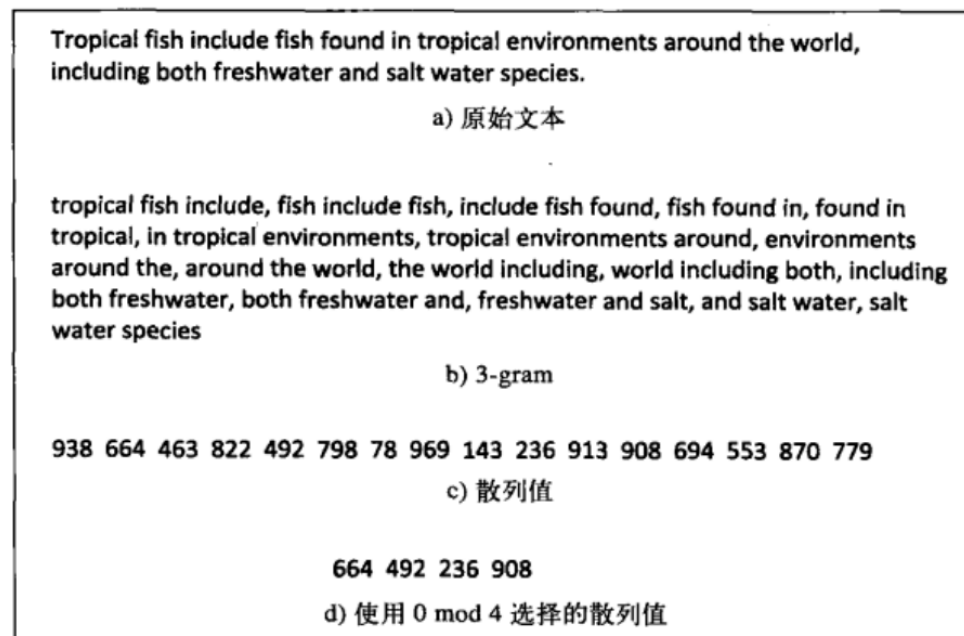
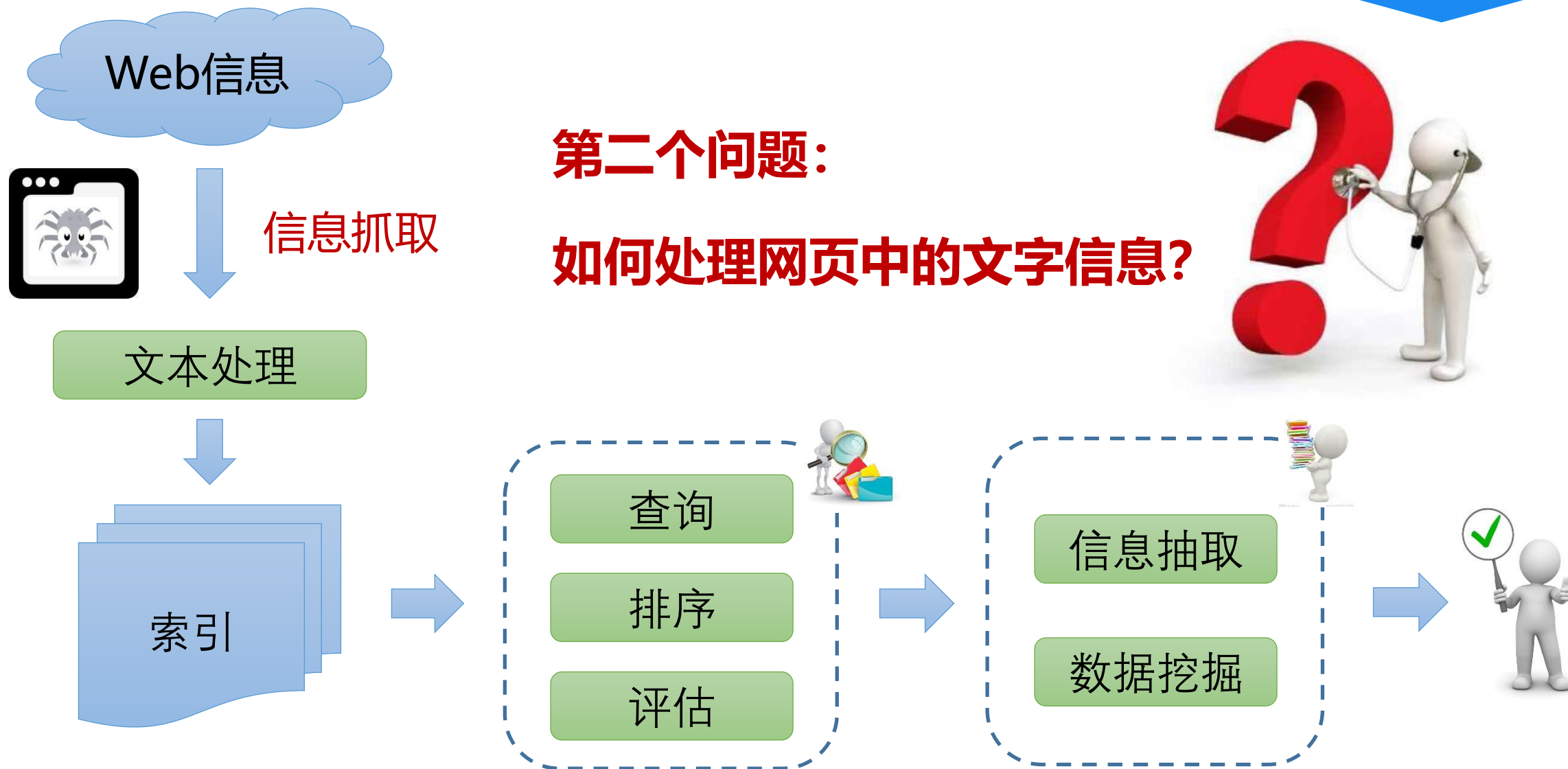


图3-14 指纹生成过程实例

- 本课程所要解决的问题



- 有效解读文档元素，是实现文档检索的第一步

**中国科学技术大学**（英语：University of Science and Technology of China，缩写：USTC），标准简称为**中国科大**，常用简称**科大**、**中科大**、**中国科技大学**或**科技大**<sup>[注 1]</sup>是**中国大陆**的一所**公立**研究型大学，主体部分文革时期从**北京**迁出，现位于**安徽省合肥市**。

**中华人民共和国**理工科排名前列的高校之一，现时属于“**泰晤士高等教育世界百强大学**”及“**软科世界百强大学**”，是“**双一流A类**”和原“**985工程**”、原“**211工程**”重点建设大学，隶属于**中国科学院**，是一所由**中国科学院**直属管理的全国**重点大学**。同时是**九校联盟**（C9）和**长三角高校合作联盟**成员，**中国大学校长联谊会**成员。2017年被教育部评选为**世界一流大学**。为**东亚研究型大学协会**和**环太平洋大学联盟**的成员。

中国科大实行“全院办校，所系结合”的办校方针，依托中国科学院的各个研究所，促进了教学与科研的一体化。<sup>[1]</sup>注重数理基础教学，自建校始长期实行五年制本科学制（与清华改革同步，1996年前后，商学院等少数几个院系本科学制改为四年。1995年招生的少年班预科其实也是1996年通过普通高考获得大学学籍，于2000年毕业。陈至立主导的高等教育大规模扩招施行后，2000年入学的本科新生学制全部改为四年制）。中科大是建国后成立的大学中第一批设立**研究生院**的大学；<sup>[2]</sup>还于1978年在李政道等人的主持下，创办了著名的**少年班**，80年代后期以后，逐渐演化为从高一不满16岁经历普通高考（北京高考的情况，1995年分数线大约750分之600分）招生，培养了千人上下的14岁前后入校的大学生。从1963年开始有第一届毕业生以来，科大已有72名毕业生（截至2017年）陆续当选**中国科学院**或**中国工程院院士**，约每1000名本科毕业生就产生1名院士、700多名硕士博士。<sup>[3]</sup>

From: <http://zh.wikipedia.org>

- **文本处理：概念与目的**
- 信息检索的基本组件，为后续应用（并不限于搜索）提供支撑。
  - 将原始文档转化为词项，以建立索引
  - 使面向查询条件的精准的文档匹配成为可能
- 文档处理与查询解析是相辅相成的。



- 词条化处理
  - 分词的概念与挑战
  - 常见分词方法
  - 常用分词工具
- 停用词处理
- 规范化处理

- **词条化 (Tokenization)**

- 将给定的字符序列拆分成一系列子序列的过程
  - 其中，每个子序列被称为一个词条 (Token)

输入：Friends, Romans, Countrymen, lend me your ears;

输出：

Friends	Romans	Countrymen	lend	me	your	ears
---------	--------	------------	------	----	------	------

- 词条化的主要任务就是确定正确的词条，并避免标点等因素干扰。



- 英文分词的挑战
- 词与词组的切分
  - To be or not to be...
- 标点符号的影响
  - 连字符: Self-motivation, 引号: 大鲨鱼奥尼尔 (O'Neal)
- 专有名词的拆分
  - New York University or New / York University?



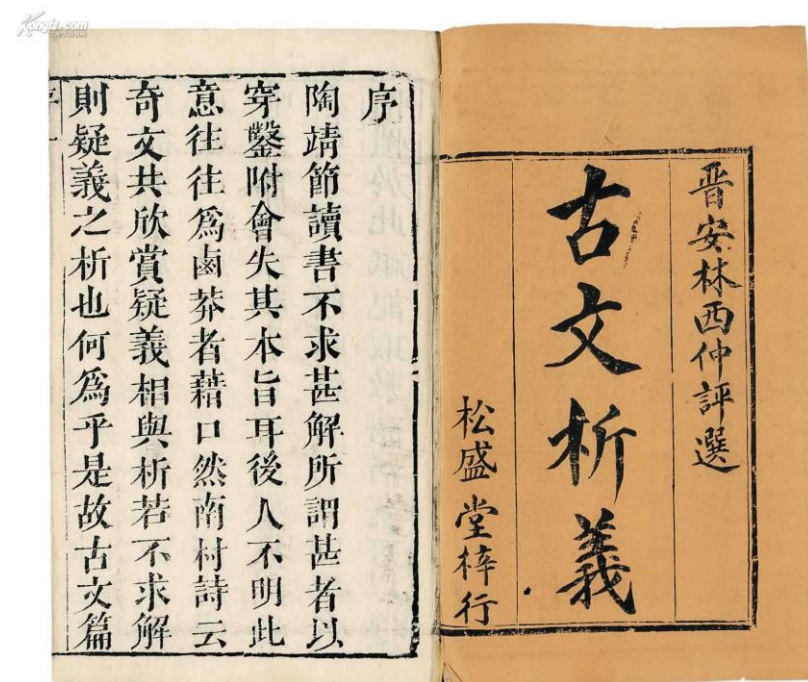


- **中文分词的挑战**

- **语素** 是最小的语音语义结合体，是最小的语言单位。
- **“字”**：简单高效，表示能力较差，不能独立地完整地表达语义信息。
  - 国家标准GB2312-80 中定义的常用汉字为6763个。
- **“词”**：具有固定的语音形式，可以独立运用的最小的语言单位。
  - 词的表示能力较强，但汉语词的个数在10万个以上，面临复杂分词问题。



- 中文分词的挑战
- 最大的挑战：没有显式分隔符（如空格）
  - 英语可视作词的集合，而汉语则是字的集合
  - 无显式分隔符使分辨不同组合方式更加困难
- 中文对虚词的运用：不单独表意，但影响句意
  - 古虚词：之乎者也，现代虚词：的、了、吧...
- 分词歧义、未登录词等



- 常见的三类中文分词歧义

- 交际

- -

- 组成

- -

- 真假

- 

你们货拉拉拉不拉拉布拉多？

知识  
盲区

+

货拉拉拉不拉拉布拉多取决于  
货拉拉在拉拉不拉多时拉布拉  
多拉的多不多

15.2w



- 未登录词的影响（中英皆然）
- 人名、地名、机构名、商品名等专有名词
  - 例如，肯尼迪（陆） / 甘乃迪（台）
- 专业领域的大量术语

Sample:

`{"originalText": "2014-08因食纳差，检查发现下腹包块，2014-09-03在全麻上行剖腹探查术，术中见肿瘤位于十二指肠，约10*11*10CM，质硬，与周围组织粘连致密无法切除，行胃空肠吻合术（未见手术记录）。术后病理：（十二指肠肿物）梭形细胞肿瘤，瘤细胞细胞增生活跃。"`

- 大量涌现的新词语、变异词语
  - 例如，“泰裤辣”、“比博燃”、“绝绝子”、“只因”



- 其他类型的字符序列
- 专业术语中文字与符号结合的部分
  - 例如, C++、B-52
- 新类型的字符序列
  - 例如, 电子邮箱地址、URL、快递单号等
- 多种语言混杂的表达方式
  - 例如, yyds、awsl、wdnmd



- 词条化处理

- 分词的概念与挑战
- 常见分词方法
- 常用分词工具

- 停用词处理

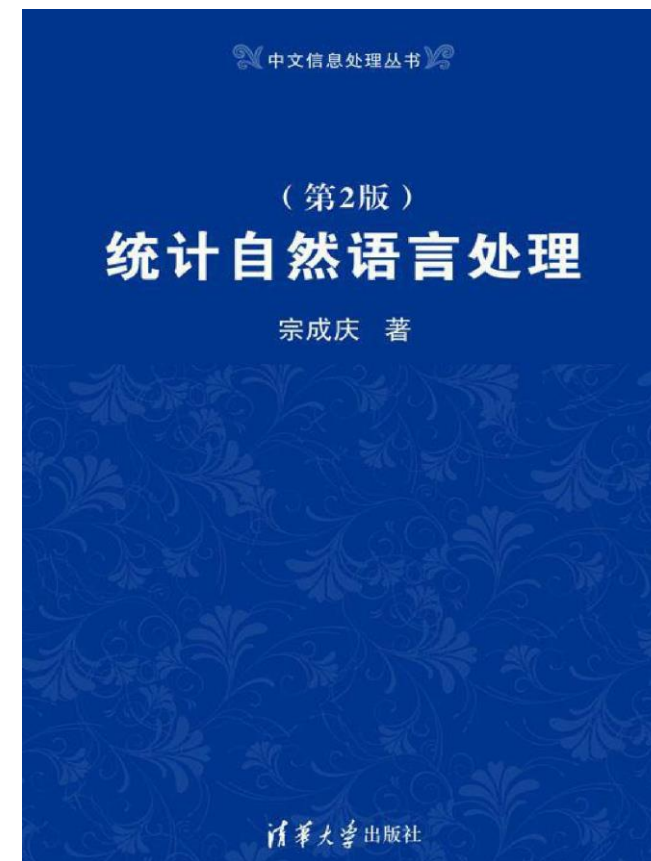
- 规范化处理

- **推荐参考书籍**

- 统计自然语言处理

- 作者: 宗成庆
- 出版社: 清华大学出版社
- 出版年: 2008-5

- 其中，第六章介绍了与统计分词方法有关的概率图模型技术，而7.1-7.2节介绍了与自动分词相关的问题背景、挑战与基本的统计方法。



- 常用的分词方法

- 基于字符匹配的方法

- 最大匹配分词法
- 最少切分分词法

- 基于统计的分词方法

- N-gram及其变形
- HMM与CRF
- 基于深度学习的分词方法

- 拓展阅读资料：近年有关中文分词的进展, <https://zhuanlan.zhihu.com/p/53521380>



- **基于匹配的分词方法**

- 又称机械分词方法，它按照一定的策略将待分析的汉字串与一个 **“充分大的”** **机器词典**中的词条进行匹配，若在词典中找到某个字符串，则匹配成功。
- 常用的机械分词方法
  - 正向最大匹配分词 (FMM)
  - 反向最大匹配分词 (RMM)
  - 双向最大匹配分词 (BM: FMM+RMM)
  - 最少切分分词 (最短路径分词)



- 基于匹配分词的一般模型

- 对于机械分词，可以建立一般模型，其形式化表达如下：
  - $ASM(d,a,m)$ ，即Automatic Segmentation Model。其中：
    - $d$ ，表示匹配方向，+1为正向，-1为逆向
    - $a$ ：每次匹配失败后增/减字符数，+1为增字，-1为减字
    - $m$ ：最大/最小匹配表示，+1为最大匹配，-1为最小匹配
  - 例如， $ASM(+, -, +)$  即正向减字最大匹配（即FMM方法）
  - 对于现代汉语而言，最大匹配更为实用（最小匹配过于琐碎）

- **正向最大匹配分词**

- Forward Maximum Matching method, FMM。
- 从左至右尽可能查找最长的词，直到当前字符与已经处理的字符串不构成词，输出已经识别的词，并从识别出来的词后面接着继续查找下一个词。
- 分词速度较快，但错误率较高（约1/169）。

例1：“使用户满意”



使用 / 户 / 满意

例2：“只有在市场中国有企业才能发展”



只有 / 在 / 市场 / 中国 / 有 / 企业 / 才能 / 发展



- 反向最大匹配分词

- Reverse (也作Backward) Maximum Matching method, RMM。
- 从右至左尽可能查找最长的词，直到当前字符与已经处理的字符串不构成词。
- 统计证实RMM分词效果更好（错误率约1/245）。为什么？

例：“使用户满意”



使 / 用户 / 满意

例：“只有在市场中国有企业才能发展”



只有 / 在 / 市场 / 中 / 国有 / 企业 / 才能 / 发展



## • 双向最大匹配分词

- Bi-directional Matching method, BM。
- 综合比较FMM与RMM两种方法的切分效果，从而选择正确的切分。
- 有助于识别分词中的交叉歧义。

例：“南京市长江大桥”  南京市 / 长江大桥 (BMM)  
  
南京市长 / 江 / 大桥 (FMM)

直接合并：南京市，长江大桥，南京市长，江，大桥

词数最少：南京市，长江大桥

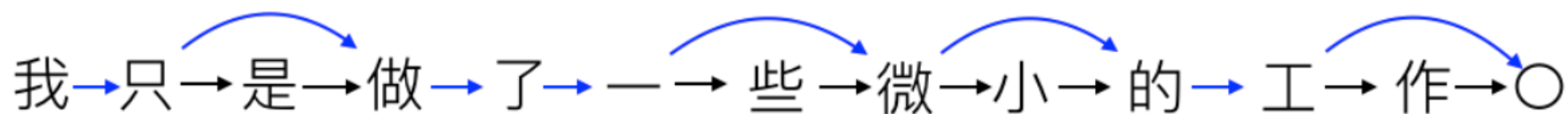


- **双向最大匹配分词**

- 关于双向最大匹配分词，一些有趣的数据
  - 经研究表明，90%的中文使用正向最大匹配分词和逆向最大匹配分词能得到相同的结果，而且保证分词正确
  - 9%的句子是正向最大匹配分词和逆向最大匹配分词切分有分歧的，但是其中一定有一个是正确的
  - 不到1%的句子是正向和逆向同时犯相同的错误：给出相同的结果但都是错的。
- 另一个有趣的统计：在随机挑选的3680个句子中，正向匹配错误而逆向匹配正确的句子占比9.24%，正向匹配正确而逆向匹配错误的情况则没有被统计到

- **最少切分分词方法**

- 使句子中切出的词数目最少。
- 等价于在有向图中搜索最短路径的问题。
  - 将每个字元视作节点，每个词形成一条边。
  - 边权重可都视为1，也可根据词频决定（尽量切出高频词）
    - 结合权重/概率之后，实际上可视为基于统计的分词方法



# 最少切分分词方法

- 拓展方法：N-最短路径法，保留N条最短的路径，以提供更多分词方案。

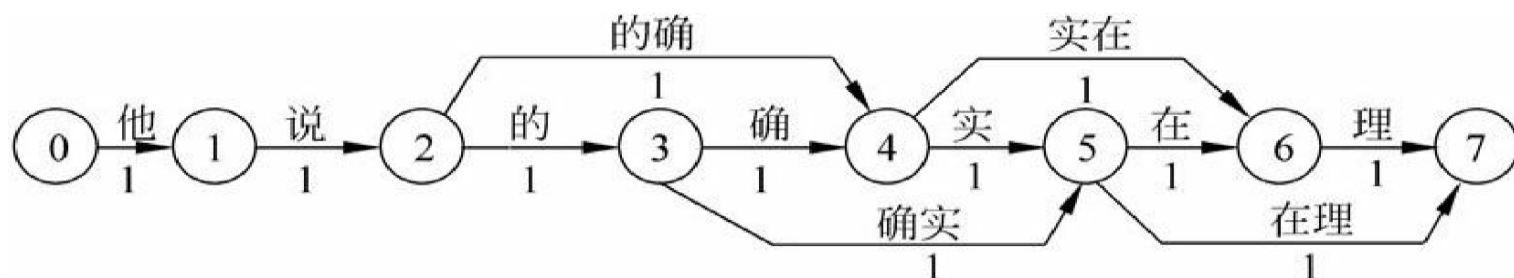


Table (4)

编号	长度	前驱
1	3	(2,1)
2	4	(3,1)

Table (5)

编号	长度	前驱
1	4	(3,1) (4,1)
2	5	(4,2)

Table (6)

编号	长度	前驱
1	4	(4,1)
2	5	(5,1)
3	6	(5,2)

Table (7)

编号	长度	前驱
1	5	(5,1) (6,1)
2	6	(5,2) (6,2)
3	7	(6,3)

Table (2)

(1,0)
-------

Table (3)

(2,0)
-------

拓展思考：N-最短路径法的复杂度是多少？



- **基于匹配分词方法的优劣**
- 优点：效率高、直观性好
- 缺点：对词典的**依赖性**
  - 维护高质量词典需要极大的开支
  - 永远难以应对新生词汇
  - 词汇频率/重要性往往对结果不产生影响

- **基于统计的分词方法**

- 没有词典，怎么办？从海量文档中去找答案。
- 字与字相邻共现的频率或概率能够较好的反映成词的可信度。
- 如果某两个词的组合在统计上出现的几率非常大，那么我们就认为分词正确。
  - 例如，“上海市长江大桥”。
  - 统计显示，“上海市 / 长江大桥”同时出现的概率，大于“上海市长 / 江 / 大桥”的概率。
  - 那么，“上海市 / 长江大桥”是正确分词的可能性更大。

- 统计分词的形式化表达

- $c = c_1 c_2 \dots c_n$ ,  $c$ 是待分词的句子（字串）。而 $w = w_1 w_2 \dots w_m$ 是切分的结果。
- 设 $P(w|c)$ 为 $c$ 切分为 $w$ 的某种估计概率。
- $w_a, w_b, \dots, w_k$ 为 $c$ 的所有可能的分词方案。
- 那么，基于统计的分词模型就是找到目标词串 $w$ ，使得 $w$ 满足：
  - $P(w|c) = \max\{P(w_a|c), P(w_b|c), \dots, P(w_k|c)\}$
  - 即估计概率最大所对应的词串。

- **统计分词的一般化过程**

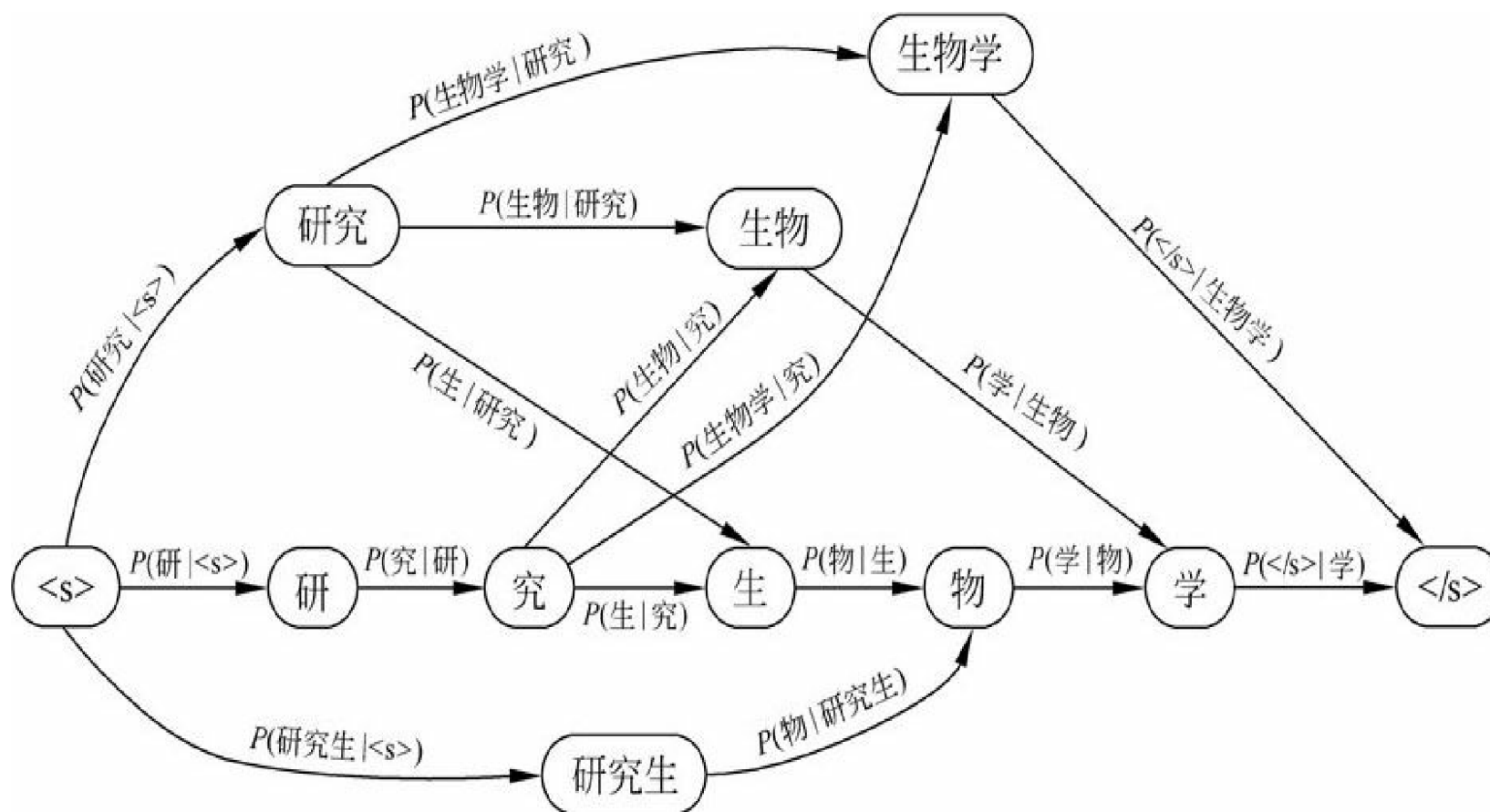
1. 建立统计语言模型
  2. 对句子按不同方案进行分词
  3. 计算不同分词方案的概率，选出概率最大的分词结果
- 理论上，基于统计的分词方法可以不需要词典，但实际应用中第2步可以采用机械分词方法进行分词，以获得候选的分词集合。
    - 既发挥匹配分词切分速度快、效率高的特点。
    - 又利用了无词典分词结合上下文识别生词、自动消除歧义的优点。

- **N-gram模型与马尔科夫假设**

- N-gram指一个由N个单词组成的集合，各单词具有先后顺序。
- N-gram模型的马尔可夫假设：
  - 当前状态出现的概率仅同过去有限的历史状态有关，而与其他状态无关。
  - 具体到分词任务，就是文本中第N个词出现的概率仅仅依赖于它前面的N-1个词，而与其他词无关。
- 常见的N-gram模型：
  - $N = 1$ ，一元语法模型（最大概率模型）， $P(w) = P(w_1) P(w_2) \dots P(w_n)$
  - $N = 2$ ，Bigram模型， $P(w) = P(w_1) P(w_2|w_1) \dots P(w_n|w_{n-1})$
  - $N = 3$ ，Trigram模型， $P(w) = P(w_1) P(w_2|w_1) P(w_3|w_1 w_2) \dots P(w_n|w_{n-2} w_{n-1})$

- N-gram模型与马尔科夫假设

- 二元文法 (Bigram) 模型 ( $N = 2$ ) 的实例



- **N-gram模型的概率估计**

- 以Bigram模型为例，基于最大似然估计进行推断

$$\text{Bigram: } P(w_n | w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

- 其中,  $C(w_{n-1}w_n)$ 指词序列 $w_{n-1}w_n$ 在语料库中出现的次数。
- 而 $C(w_n)$ 指某个单词 $w_n$ 在语料库中出现的次数。

- **N-gram模型的分词过程**

- 以Bigram模型为例

1. 首先，构造训练语料库，计算所有的 $C(w_n)$ 与 $C(w_{n-1}w_n)$ 。

2. 其次，对于每一个可能的分词序列 $w$ ，计算以下公式

- $$P(w) = P(w_1) P(w_2|w_1) \dots P(w_n|w_{n-1})$$

- 其中， 
$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

3. 最后，返回最大的 $P(w)$ 所对应的分词序列作为结果。



- **Bigram模型实例**

- 例如，判断句子“我想晚上去吃意大利菜”的分词。
- 已知总词数为13,748，各个词出现的频率为：

我	想	晚上	去	吃	意大利	菜
3437	1215	3256	938	213	1506	459

- 同时，各个词序列的共现次数如下：

	我	想	晚上	去	吃	意大利	菜
我	8	1087	0	13	0	0	0
想	3	0	786	0	6	8	6
晚上	3	0	10	860	3	0	12
去	0	0	2	0	19	2	52
吃	2	0	0	0	0	120	1
意大利	19	0	17	0	0	0	0
菜	4	0	0	0	0	1	0

- **Bigram模型实例**

- 基于上述统计数据，计算 “我 / 想 / 晚上 / 去 / 吃 / 菜” 这一分词的可能性。
- $P(\text{我}) * P(\text{想} | \text{我}) * P(\text{晚上} | \text{想}) * P(\text{去} | \text{晚上}) * P(\text{吃} | \text{去}) * P(\text{菜} | \text{吃})$
- $3437/13748 * 1087/3437 * 786/1215 * 860/3256 * 19/938 * 1/213 = 0.00000128$



- 特殊形式：一元语法模型

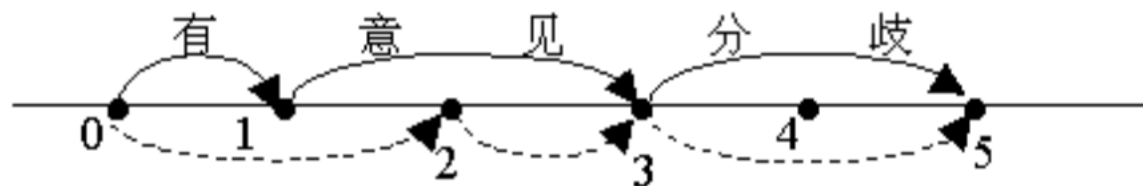
- 当 $N=1$ 时，N-gram模型退化为一元语法模型，此时词与词之间是独立的。

$$P(W) = P(w_1, w_2, \dots, w_i) \approx P(w_1) \times P(w_2) \times \dots \times P(w_i)$$

独立性假设，一元语法

$$P(w_i) = \frac{w_i \text{在语料库中的出现次数 } n}{\text{语料库中的总词数 } N}$$

- 例如，判断句子“有意见分歧”的分词方案。



- 一元语法模型实例

- 待分词句子：“有意见分歧”

- W1: 有/ 意见/ 分歧/
- W2: 有意/ 见/ 分歧/

- $P(W1) = P(\text{有}) * P(\text{意见}) * P(\text{分歧})$   
 $= 1.8 \times 10^{-9}$

- $P(W2) = P(\text{有意}) * P(\text{见}) * P(\text{分歧})$   
 $= 1 \times 10^{-11}$

- 由于 $P(W1) > P(W2)$ ，因此，第一种分词更合理。

词语	概率
...	...
有	0.0180
有意	0.0005
意见	0.0010
见	0.0002
分歧	0.0001
...	...

- **基于统计文法模型的优劣**

- 优点：减轻了对于词典的依赖性
  - 然而，这种依赖并非完全消除，取决于性能与效率的平衡
    - 如果深度结合机械分词（匹配分词），则效率提升但依赖词典
    - 如果减少对词典的依赖，则需要更多地遍历潜在的组合（解空间巨大！）
- 缺点：依赖已有数据中词频的统计，对于新生词汇或专业词汇不友好
  - 冷门领域的稀有词汇往往难以准确划分
  - 易受数据集先验偏差（Bias）的影响

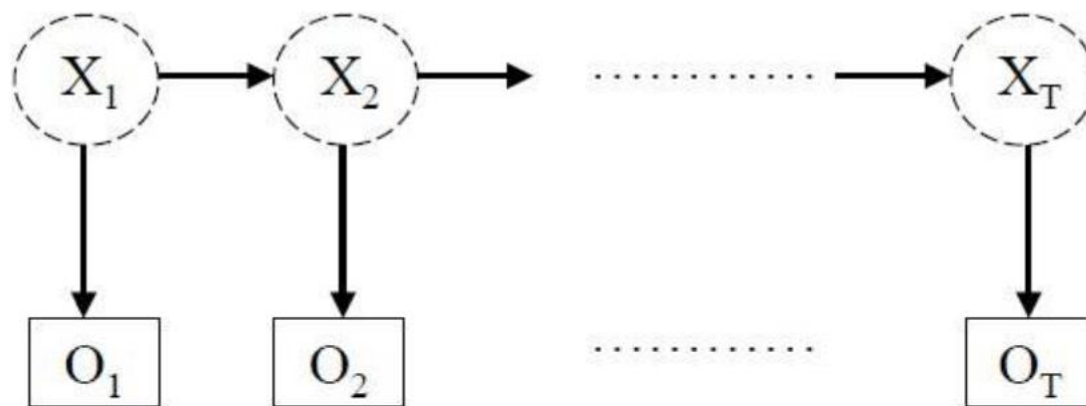
- **基于序列标注的分词方法**

- 基于词典的机械分词方法所面对的主要问题在于未登录词识别。
- 基于统计模型的分词方法，进一步抽象而言，可以得到一个序列标注问题
  - 四类标注：B（词的开始）、M（词的中间）、E（词的结束）、S（单字词）
  - 例子：中国科学技术大学是中国最好的大学
    - 标注：BMMMMMME S BE BME BE
    - 分词结果：中国科学技术大学 / 是 / 中国 / 最好的 / 大学

- **隐马尔可夫模型 (HMM)**

- 隐马尔可夫模型 (HiddenMarkov Model)

- 基本的思想是根据观测值序列，找到真正的隐藏状态值序列
  - 在中文分词中，每个字符是观测值，而标签 (BMES) 为隐藏状态值



- **隐马尔可夫模型 (HMM)**

- 隐马尔可夫模型的五个核心要素：两个集合、三个矩阵
  - 两个集合：观测值集合（字符集合）、隐藏状态值集合（BEMS）
  - 三个矩阵：
    - 初始状态概率矩阵：第一个字属于某种隐含状态（BMES）的概率
    - 隐含状态转移概率矩阵：各种隐含状态（各种标签）之间的转移概率
    - 观测状态概率矩阵：从隐含状态（标签）到观测值（字符）的转移概率



- **隐马尔可夫模型 (HMM)**

- 根据上述元素，我们将隐马尔可夫模型下的中文分词问题表述为如下形式：
- 当我们观测到句子 $w_1, w_2 \dots w_n$ ，其中 $w_i$ 为第 $i$ 个汉字，我们希望找到相应的标签序列 $s_1, s_2 \dots s_n$ ，其中 $s_i$ 为 $w_i$ 对应的标签（BMES中的一种），使得 $P(s_1, s_2 \dots s_n | w_1, w_2 \dots w_n)$ 概率最大。
- 为求解这一目标函数，需要隐马尔可夫模型的两个基本假设：
  - 齐次假设：当前隐藏状态只与上一个状态有关系
    - 即 $P(s_i | s_{i-1}, s_{i-2} \dots s_1) = P(s_i | s_{i-1})$
  - 观测独立性假设：观测值之间互相独立的，只与生成它的状态有关系
    - 即 $P(w_1, w_2 \dots w_n | s_1, s_2 \dots s_n) = P(w_1 | s_1)P(w_2 | s_2) \dots P(w_n | s_n)$

- **隐马尔可夫模型 (HMM)**

- 基于目标函数和两个基本假设，基于隐马模型的中文分词问题转化为：
- 我们希望找到相应的标签序列 $s_1, s_2 \dots s_n$ ，使得 $\prod_{i=1}^N P(s_i | s_{i-1}) P(w_i | s_i)$ 概率最大。
- 针对这一问题，可采用**维特比 (Viterbi) 算法**进行求解：
  1. **初始化**：对第一个字，分别以BEMS四种状态计算其概率
  2. **递归**：对第 $i$ 个字，遍历四种状态，先计算该状态最可能是由前一时刻的哪个状态转换而来的，再乘以该状态下得到观测值（字）的概率，取最大值。
  3. **终止**：在第 $n$ 个字时，取得到的最大概率，并得到最后一个字的状态标签。
  4. **回溯**：由最优路径的终点向前，找到各个时刻的最优状态，还原全部标签。

## • 隐马尔可夫模型 (HMM)

- 例题：记 $S$ 是所有可能的状态的集合， $S = \{s_1, s_2\} = \{T, F\}$
- $I = (s_{i_1}, s_{i_2}, s_{i_3})$ 是长度为3的状态序列，其对应的观测序列为 $W = (w_1, w_2, w_3) = (X, Y, Z)$
- 隐含状态转移概率矩阵为  $A = [a_{ij}]_{2 \times 2} = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$ ，其中， $a_{ij} = P(i_{t+1} = s_j | i_t = s_i) \quad i=1,2; j=1,2$   
表示在时刻  $t$  处于状态  $s_i$  的条件下在时刻  $t+1$  转移到状态  $s_j$  的概率。
- 观测状态概率矩阵为  $B = [b_j(k)]_{2 \times 3} = \begin{pmatrix} 0.5 & 0.4 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}$ ，其中， $b_j(k) = P(w_t = v_k | i_t = s_j) \quad k=1,2,3; j=1,2$   
表示在时刻  $t$  处于状态  $s_j$  的条件下生成观测  $v_k$  的概率。
- $\pi$  是初始状态概率向量： $\pi = (\pi_1, \pi_2) = (0.6, 0.4)$ ，其中， $\pi_i = P(i_1 = s_i) \quad i=1,2$   
表示在时刻  $t=1$  处于状态  $s_i$  的概率

- **隐马尔可夫模型 (HMM)**

- 前述例题中对应的三个矩阵:

初始状态概率矩阵

T	F
0.6	0.4

隐含状态转移概率矩阵

	->T	->F
T	0.7	0.3
F	0.4	0.6

观测状态概率矩阵

	->X	->Y	->Z
T	0.5	0.4	0.1
F	0.1	0.3	0.6

问：观测到显式状态序列依次为X、Y、Z，最可能的隐含状态序列是什么？

- 隐马尔可夫模型 (HMM)

### (1) 初始化

在  $t=1$  时, 对每一个状态  $s_i$  (记显式状态  $T$  为  $s_1$ ,  $F$  为  $s_2$ ) , 计算状态为  $s_i$  观测  $w_1$  为  $x$  的概率, 记此概率为  $\delta_1(s_i)$  , 则

$$\delta_1(s_i) = \pi_i b_i(w_1) \quad i=1,2$$

代入实际数据

$$\delta_1(T) = 0.6 \times 0.5 = 0.3$$

$$\delta_1(F) = 0.4 \times 0.1 = 0.04$$

**可见, 第一个隐状态为T更为合理**

## • 隐马尔可夫模型 (HMM)

(2) 在 $t=2$ 时, 对每一个状态 $s_i$ , 求在 $t=1$ 时状态为 $s_j$ 观测为 $X$ 并在 $t=2$ 时状态为 $s_i$ 观测 $w_2$ 为 $Y$ 的路径的最大概率, 记此最大概率为 $\delta_2(s_i)$ , 则

$$\delta_2(s_i) = \max_{1 \leq j \leq 2} \{ \delta_1(s_j) a_{ji} \} b_i(w_2)$$

同时, 对每个状态 $s_i$ , 记录概率最大路径的前一个状态 $s_j$ 所对应的下标 $j$ :

$$\psi_2(s_i) = \arg \max_{1 \leq j \leq 2} \{ \delta_1(s_j) a_{ji} \} \quad i = 1, 2$$

计算:

$$\begin{aligned} \delta_2(T) &= \max_{1 \leq j \leq 2} \{ \delta_1(s_j) a_{j1} \} b_1(w_2) \\ &= \max_j \{ 0.3 \times 0.7, 0.04 \times 0.4 \} \times 0.4 \\ &= 0.084 \end{aligned}$$

$$\delta_2(F) = 0.027$$

$$\begin{aligned} \psi_2(T) &= \arg \max_{1 \leq j \leq 2} \{ \delta_1(s_j) a_{j1} \} \\ &= \arg \max_j \{ 0.3 \times 0.7, 0.04 \times 0.4 \} \\ &= 1 \end{aligned}$$

$$\psi_2(F) = 1$$

**显然, 第二个隐状态为T的概率仍然高于F的概率**

## • 隐马尔可夫模型 (HMM)

同理，在  $t = 3$  时，

$$\begin{aligned}\delta_3(T) &= \max_{1 \leq j \leq 2} \{ \delta_2(s_j) a_{j,1} \} b_1(w_3) \\ &= \max_j \{ 0.084 \times 0.7, 0.027 \times 0.4 \} \times 0.1 \\ &= 0.00588\end{aligned}$$

$$\delta_3(F) = 0.01512$$

此时，第三个隐状态为F的概率更高

$$\begin{aligned}\psi_3(T) &= \arg \max_{1 \leq j \leq 2} \{ \delta_2(s_j) a_{j,1} \} \\ &= \arg \max_j \{ 0.084 \times 0.7, 0.027 \times 0.4 \} \\ &= 1\end{aligned}$$

$$\psi_3(F) = 1$$

(3) 以  $P^*$  表示最优路径的概率，则

$$P^* = \max_{1 \leq i \leq 2} \delta_3(s_i) = 0.01512$$

最优路径的终点下标  $i_3^*$  是  $i_3^* = \arg \max_i \{ \delta_3(s_i) \} = 2$

- 隐马尔可夫模型 (HMM)

(4) 由最终路径的终点下标  $i_3^*$ ，逆向找到  $i_1^*, i_2^*$ ：

在  $t=2$  时,  $i_2^* = \psi_3(s_{i_3^*}) = \psi_3(F) = 1$

在  $t=1$  时,  $i_1^* = \psi_2(s_{i_2^*}) = \psi_2(T) = 1$

于是求得  $(i_1^*, i_2^*, i_3^*) = (1, 1, 2)$

最优状态序列  $I^* = (s_1, s_1, s_2) = (T, T, F)$

注：上述案例可以通过维基百科词条进行回顾 [https://en.wikipedia.org/wiki/Viterbi\\_algorithm](https://en.wikipedia.org/wiki/Viterbi_algorithm)

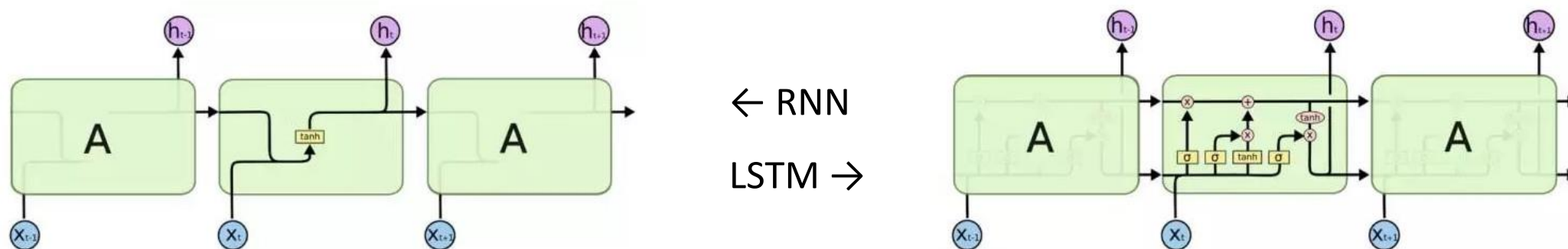


- **条件随机场模型 (CRF)**

- 隐马尔可夫模型的独立性假设难以描述字词之间的复杂关联。
- 条件随机场模型 (Conditional Random Field)
  - 具有表达长距离依赖性和交叠性特征的能力
  - 所有特征可以进行全局归一化，能够求得全局的最优解。
- 有关条件随机场的深入学习，可参见ICML 2001论文或《统计自然语言处理》第6.9节
  - Lafferty, et. al. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." ICML 2001
- 相关工具包：CRF++ ( <https://taku910.github.io/crfpp/> ) 、 Genius ( <https://github.com/duanhongyi/genius> ) 等

## • 长短时记忆模型 (LSTM)

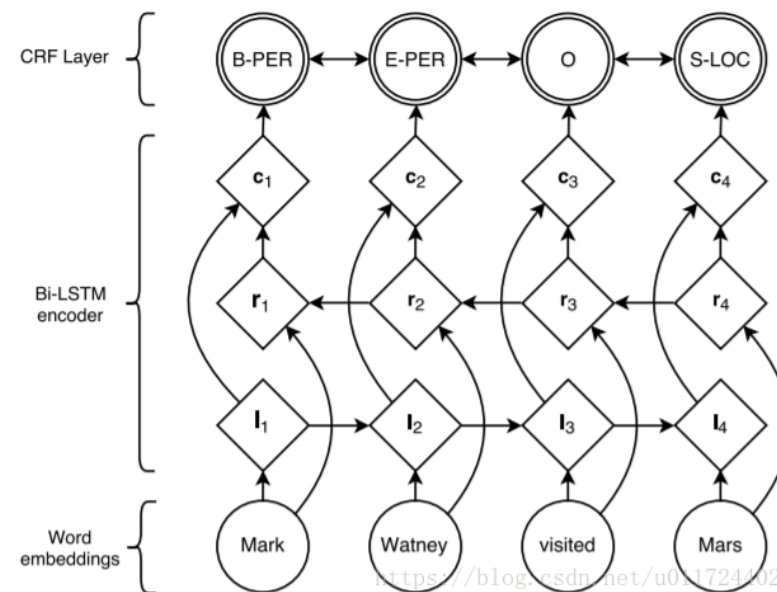
- 传统的神经网络中，每层内部的节点之间是无连接的，因此无法利用上下文关系。而循环神经网络 (Recurrent Neural Networks) 则有效解决了这一问题。
- 由于语句长短不同，当需要较长上下文关系时，RNN对信息依赖的学习能力有限。



- 长短时记忆模型 (Long Short-Term Memory) 通过四层神经网络代替RNN中原有的单一神经网络层，使其拥有增加或减少信息的能力。
  - 新增了保存长期信息的单元状态，以及控制保存、输出长期状态和输入瞬时状态的“门”

## • LSTM与CRF的结合

- 基于LSTM方法解决分词问题，传统解决思路是采用LSTM+SoftMax分类的思路。
  - 忽略了预测序列的标签之间的关联性。
  - 可能导致错误标签序列的出现，例如：B后又出现了B。
- 通过结合LSTM与CRF技术，将有效利用句子级别的标记信息。
  - 输出的将不再是相互独立的标签，而是最佳且合理的标签序列。



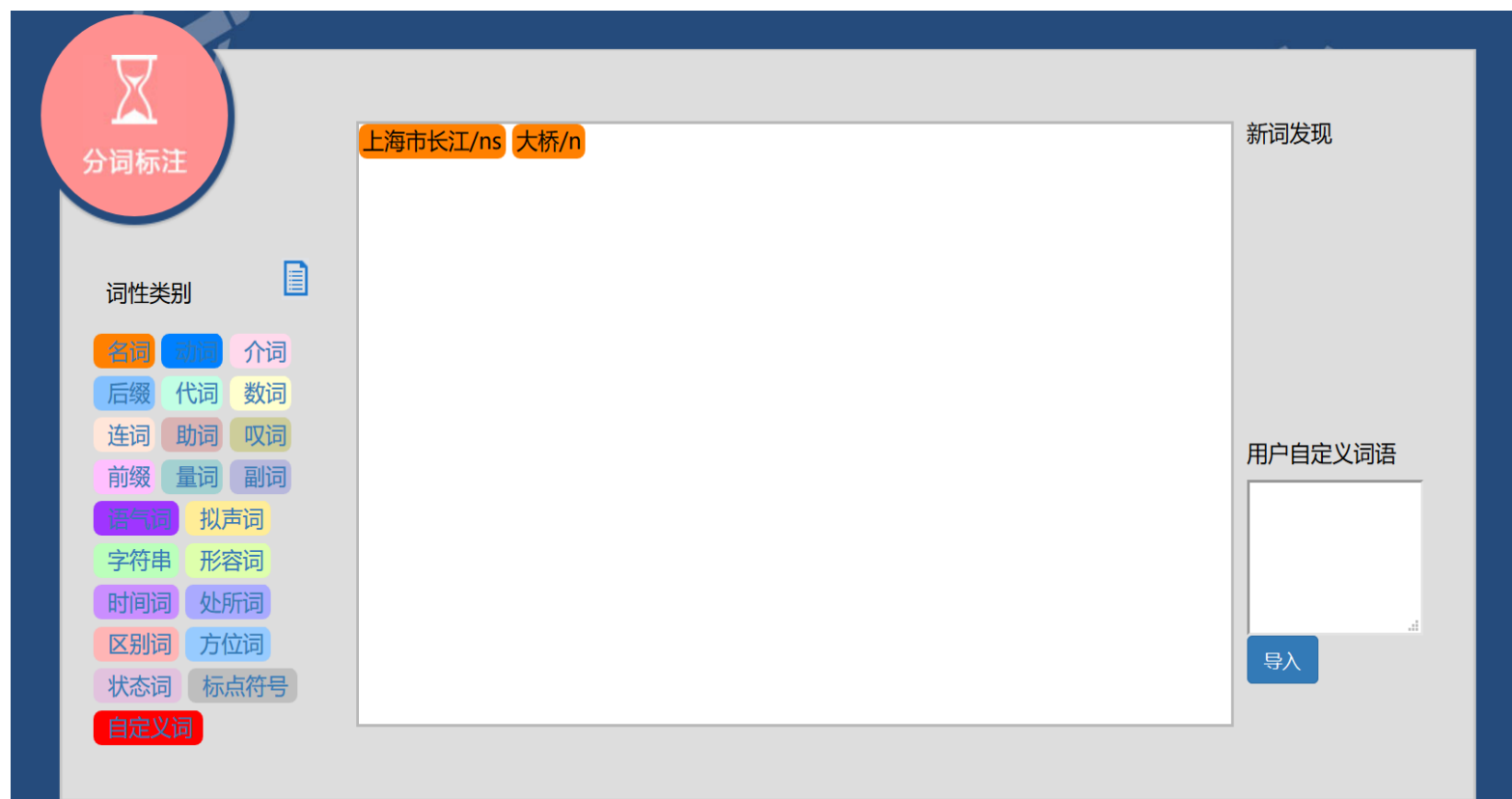
- 词条化处理

- 分词的概念与挑战
- 常见分词方法
- 常用分词工具

- 停用词处理

- 规范化处理

- 常用的中文分词工具
- **NLPIR-ICTCLAS**: <https://github.com/NLPIR-team/NLPIR>
- 中科院/北理工研发，基于HMM技术，界面友好，效果尚可。



- **常用的中文分词工具**
- **结巴分词**: <https://github.com/yanyiwu/cppjieba/>
- 基于HMM技术, 有专门的Python库支持: <https://github.com/fxsjy/jieba>
- 语言支持最丰富 (Java、C++、R等), 支持多种分词模式, 支持自定义词典
- **HanLP**: <https://github.com/hankcs/HanLP>
- 由大快搜索主导并完全开源, 语料时新、可自定义
- **THULAC**: <https://github.com/thunlp/THULAC>
- 集成了当时规模最大的人工分词和词性标注中文语料库, 效果好, 速度快

- **常用的中文分词工具**
- **PKUSeg**: <https://github.com/lancopku/PKUSeg-python>
- 北大研发，分词准确率大幅提升。
- 支持多领域分词，支持用全新的标注数据来训练模型。

MSRA	F-score	Error Rate
jieba	81.45	18.55
THULAC	85.48	14.52
pkuseg	<b>96.75 (+13.18%)</b>	<b>3.25 (-77.62%)</b>

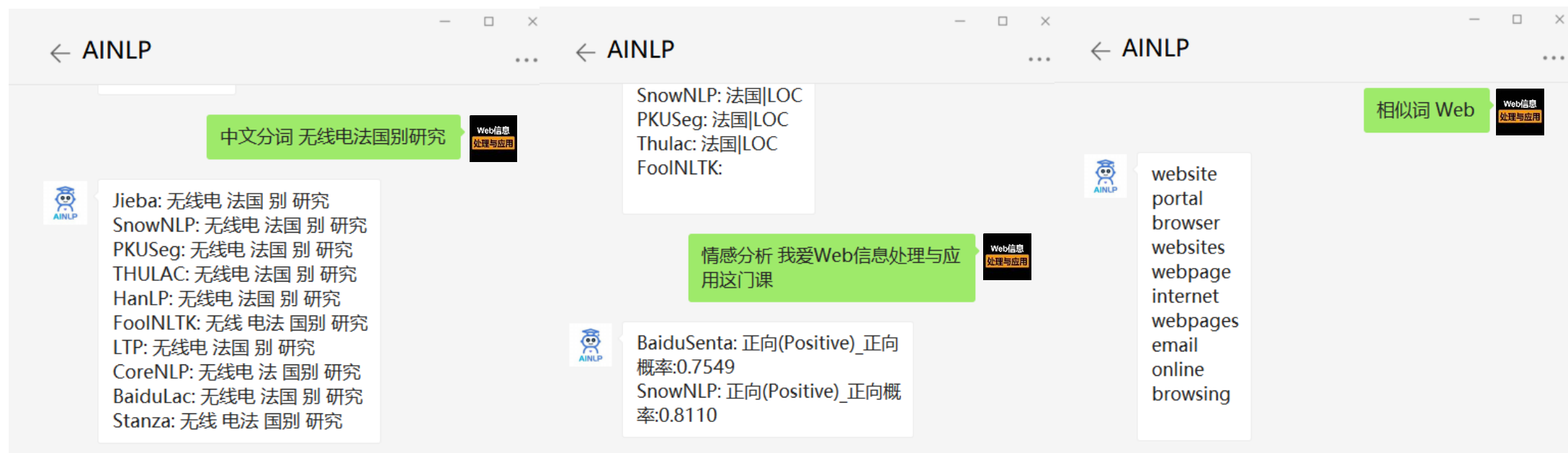
- **常用的英文分词工具**
- **Stanford NLP**: <https://nlp.stanford.edu/software/index.shtml>
- 支持多种语言的完整文本分析管道
  - 包括分词、词性标注、词形归并和依存关系解析等任务
- 提供了与 CoreNLP 的 Python 接口。
  - CoreNLP: <https://stanfordnlp.github.io/CoreNLP/>
- Stanford Word Segmenter
  - <https://nlp.stanford.edu/software/segmenter.html>



- 分词工具包测试

- 公众号: AINLP

- 支持包括Jieba、PKUSeg、THULAC等在内的10种分词工具的测试
- 还支持词性标注, 命名实体识别、情感分析, 相似词查询, 诗句生成等



- **分词可能带来的隐患**
- 分词带来的大量低频词，导致严重的数据稀疏。
- 越来越多的OOV（Out of Vocabulary）词。
- 分词中难免的错漏将导致额外的噪声。
- 深度学习发展，分词的收益愈发有限。
- 大模型时代，语义理解工具远胜以往。



- 词条化处理
  - 分词的概念与挑战
  - 常见分词方法
  - 常用分词工具
- 停用词处理
- 规范化处理

- **停用词的概念与意义**

- 停用词, Stopwords, 指文档中频繁出现或对实际语义影响不大的词语。
  - 例如, 英文中的The、 of, 中文中的 “的” 、 “是” 等。
  - 数字、 副词等与语义关系不大的词常作为停用词被处理。
- 为什么要去除停用词?
  - 重复率很高, 会造成索引中的倒排表很长, 影响查询性能。
  - 对最后结果的排序没什么贡献, 反而可能产生干扰。

**ST****P!**

- **停用词类型与识别**

- 停用词的设置与语料库的性质有关
  - 除通用停用词表外，特定学科或领域也具有其专用的停用词。
  - 例如，URL中的www，Wikipedia中的wiki
- 常用的停用词识别方法
  - 较为成熟的停用词识别方法有：文本频率、词频统计、熵计算等。
  - 更为复杂的算法将结合统计与句法或内容分析。
- 常用的停用词表：[哈工大停用词表](#)、[百度停用词表](#)、NLTK停用词表等。

- **去除停用词可能导致的隐患**
- 有些停用词在特定场景下是有意义的
  - 例如，“非”、“不”表示否定；“较”、“稍微”表示程度等。
- 有些停用词的组合是有意义的
  - 例如，“的士”、“To be or not to be”。
  - 依赖于分词的效果。

**ST****P!**

- **未来停用词的使用趋势**

- 现代搜索引擎的趋势是逐渐减少对停用词的使用。
- 现代搜索引擎更关注利用语言的统计特性来处理常见词问题。
  - 采用压缩技术，降低停用词表的存储开支。
  - 引入词项权重，将高频词的影响降至最低。
  - 索引去除技术，低于权重的词项将被排除。

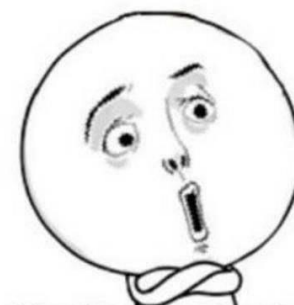
ST  P!

- 词条化处理
- 停用词处理
- **规范化处理**
  - 归一化处理
  - 错误拼写检查
  - 同义词/相关词处理



- **文本规范化的意义**

- 文本处理的主要目标在于优化查询词与索引词之间的匹配。
- 然而，两方的文本内容都可能出现各种各样的干扰情况：
  - 大小写、缩写、标点等的干扰， e.g., USA与U.S.A
  - 不同时态等导致的词形不同， e.g., have / has
  - 同义词 / 相关词等的干扰， e.g., 中科大与中国科大
  - 用户个性化表述方式， 如方言
- 规范化的目的就在于尽量保证索引词项符合用户查询输入。



讲到好似真噶甘

- **规范化需要考虑的问题**

- 大小写、标点符号、缩写等规则化处理。
- 词根化处理：词干提取与词形还原。
- 拼写错误检查与修正。
- 同义词 / 相关词识别与处理。
- 其他类型的文本规范化问题。
  - 特殊文本形式，如中文中的9月19日与英文中的9/19。
  - 跨语言问题，如日语中的假名汉字与中文中的汉字，e.g., “邪魔”。

- 词条化处理
- 停用词处理
- **规范化处理**
  - 归一化处理
  - 错误拼写检查
  - 同义词/相关词处理

- 归一化处理

- 归一化/词根化，指还原词语的特殊形式的过程。

- 例如：

- Ran, running      ➡      run

- Universities      ➡      university

- 往往针对英语等语言，汉语并不需要这一步。

- 词根化处理可以有效降低词项的数量并减少歧义。



- 词干提取

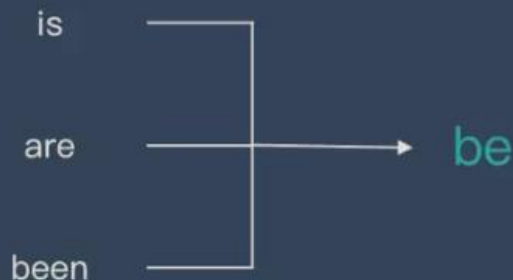
- Stemming, 指去除单词前后缀, 获得词根的过程。
  - 常见的前后词缀有 “复数形式”、“过去分词”、“进行时” 等。



- 词形还原

- Lemmatisation, 指基于词典, 将单词的复杂形态转变成最基础的形态。
  - 并不是简单地将前后缀去掉, 而是会根据词典将单词进行转换。

词形还原 – Lemmatisation



- **词干提取与词形还原的异同**

- 词干提取与词形还原的相同点

- 目标一致。两者的目标均为将词的不同形态简化或归并为基础形式。
- 结果交叠。两者不是互斥关系，其结果有部分交叉。
- 方法类似。目前两者主流方法均是利用语言中的规则或词典实现。

- 词干提取与词形还原的不同点

- 在原理上，词干提取采用“缩减”，而词形还原采用“转变”。
- 在复杂性上，词形还原需考虑词缀转化、词性识别等，更为复杂。
- 在实现上，词干提取主要利用规则变化，而词形还原更依赖于词典。
- 在结果上，词干提取不一定得到完整单词，而词形还原是完整单词。

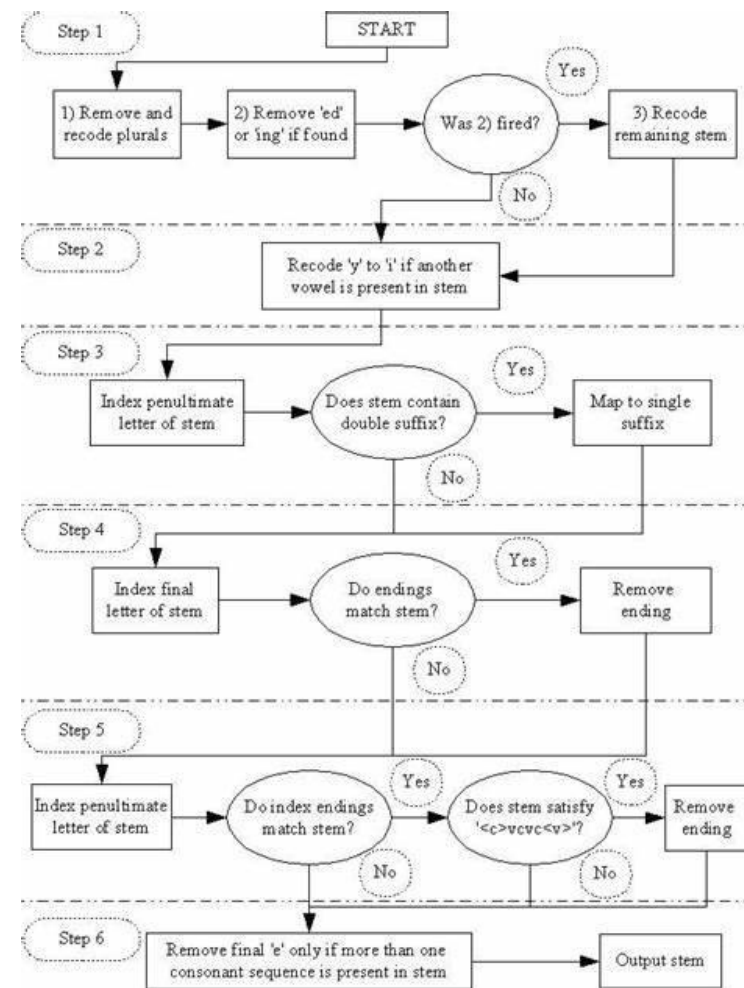
- **基本词干提取方法：Porter Stemming**

- 英文中最常用的词干提取方法。
- 使用一系列后缀变换规则对单词进行变换。
- 其开源版本可通过网络获得。
  - 例如：<http://tartarus.org/~martin/PorterStemmer/>
  - 升级版本：<http://snowball.tartarus.org/algorithms/english/stemmer.html>
  - 也可以通过以下网站进行在线简单测试：
  - <https://textanalysisonline.com/nltk-porter-stemmer>



## • Porter Stemming的基本流程

1. 去除单词的复数形式
2. 去除 -ed(ly) 或 -ing(ly) 等后缀
3. 将 -y 改为 -i
4. 处理双重后缀, 如 -ization等
5. 处理 -full, -ness等后缀
6. 处理 -ant, -ence等后缀
7. 去除掉最后的 -e和 -ll



- 词条化处理
- 停用词处理
- **规范化处理**
  - 归一化处理
  - **错误拼写检查**
  - 同义词/相关词处理

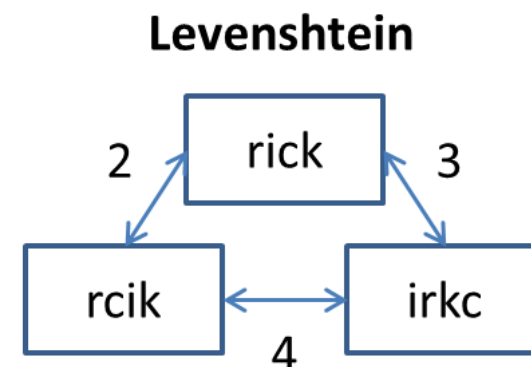
- 拼写错误处理

- 用户在输入查询条件时，往往容易出现拼写错误 (>10%)。



- **拼写错误处理**

- 通常采用基于词典或编辑距离的方式进行检查和校对。
- 编辑距离 (Levenshtein Distance)
  - 指两个字符串之间转换所最少需要的编辑操作步数。
  - 允许的一步编辑操作包括替换、插入或删除一个字符。
  - 例如:  $\text{Distance}(\text{"Kitten"}, \text{"Sitting"}) = 3$ 
    - kitten  $\rightarrow$  sitten (substitution of "s" for "k")
    - sitten  $\rightarrow$  sittin (substitution of "i" for "e")
    - sittin  $\rightarrow$  sitting (insertion of "g" at the end).



## • 拼写错误处理

## • 拼写错误处理是有极限的，不是什么错误都能够有效校正

钢铁锅含眼泪喊修瓢锅这是什么歌词

我来答 分享 举报

6个回答

#热议# 王嘉尔夹走王一博香菜，王嘉尔生活中什么性格？



苏冰堰2012 LV13  
推荐于2018-03-23

关注

你好！是《海阔天空》粤语的音译。

1. “今天我，寒夜里看雪飘过” 粤语发音音译为：钢铁锅，含眼泪喊修瓢锅。
2. 也是网上有人的恶搞翻唱。
3. 你可以去听听，希望对你有帮助。

5 评论 分享 举报



Baidu 百度

钢铁锅含眼泪喊修瓢锅



百度一下

网页 资讯 视频 图片 知道 文库 贴吧 地图 采购 更多

百度为您找到相关结果约110,000个

搜索工具

[钢铁锅,含眼泪喊修瓢锅 这是什么歌? - 知乎](#)

2016年6月26日 《海阔天空》全歌词：**钢铁锅**，**含眼泪喊修瓢锅**，坏缺烂角的换新**锅瓢**乱放。风  
雨里追锅，无泪缝把层...

知乎 百度快照

[钢铁锅,含眼泪喊修瓢锅 这是什么歌? - 百度知道](#)



40个回答 - 回答时间: 2019年10月31日

最佳答案: 《海阔天空》演唱: Beyond 今天我 寒夜里看雪飘过 怀着冷却了的心窝漂远方 风雨里追赶 雾里分不清影踪 天空海阔你与...

百度知道 百度快照

- 拼写错误处理

- 不必要的拼写错误处理将影响用户体验。



- 在判断用户真实意图的基础之上，准确理解用户输入的查询条件。
- 采用更为友好的方式处理可能的“拼写错误”。

- 词条化处理
- 停用词处理
- **规范化处理**
  - 归一化处理
  - 错误拼写检查
  - **同义词/相关词处理**

- **同义词 / 相关词处理**

- 比词根化和拼写错误更难处理，通常借助人工维护的知识库。
- 常见的词与词之间的关系：
  - 同义词, e.g., college  $\approx$  university
  - Is – a关系, e.g., Boeing 737 max is a plane.
  - Is – part – of关系, e.g., Nokia is part of Microsoft
  - 反义词, e.g., Young v.s. Old



## • 同义词 / 相关词处理

- 基于人工维护的知识库，获取各种词项之间的关系。
- 例如，WordNet: <https://wordnet.princeton.edu/>
  - 大型的英文词汇数据库，将不同词性的单词归类至不同的认知同义词集合中。
  - 目前，已有超过117000个认知同义词集合。


```
dog, domestic dog, Canis familiaris
=> canine, canid
    => carnivore
        => placental, placental mammal, eutherian, eutherian mammal
            => mammal
                => vertebrate, craniate
                    => chordate
                        => animal, animate being, beast, brute, creature, fauna
                            => ...
```


- 犬 > 类犬动物 > 食肉动物  
> 胎盘类哺乳动物 > 哺乳  
动物 > 脊椎动物...

## • 同义词 / 相关词处理

- 相关应用：基于同义词 / 相关词，拓展查询条件。

DBLP FILTER Sign in

 data model moving objects Search Sort by ☐ relevance ☒ importance ☐ date

**Scholar** About 28 results ( 5.57sec)  (1998~2013)

**Since Time**

Since 2013

Since 2012

Since 2009

Custom range...

**Sort By**

Sort By Relevance

Sort By Importance

Sort By Date

A	EE	Scholar	A Data Model and Data Structures for Moving Objects Databases. (Luca Forlizzi and Ralf Hartmut G and ü) <i>ACM Conference on Management of Data (sigmod ) [2000] Cited by 353</i>
A	EE	Scholar	Scientific Data Repositories: Designing for a Moving Target. (Etzard Stolte and Christoph von Praun and Gustavo Alonso) <i>ACM Conference on Management of Data (sigmod ) [2003] Cited by 43</i>
A	EE	Scholar	A Data Model for Moving Objects Supporting Aggregation. (Bart Kuijpers and Alejandro A. Vaisman) <i>IEEE International Conference on Data Engineering (ICDE ) [2007] Cited by 20</i>
B	EE	Scholar	Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. (Martin Erwig and Ralf Hartmut G and ü) <i>GeoInformatica (GeoInformatica ) [1999] Cited by 367</i>
B	EE	Scholar	A generic data model for moving objects. (Jianqiu Xu and Ralf Hartmut G and ü) <i>GeoInformatica (GeoInformatica ) [2013]</i>
B	EE	Scholar	An Object-Field Perspective Data Model for Moving Geographic Phenomena. (Kyoung-Sook Kim and Yasushi Kiyoki) <i>Database Systems for Advanced Applications (DASFAA ) [2010]</i>
C	EE	Scholar	Place: A Distributed Spatio-Temporal Data Stream Management System for Moving Objects. (Xiaopeng Xiong and Hicham G. Elmongui and Xiaoyong Chai) <i>International Conference on Mobile Data Management (MDM ) [2007] Cited by 18</i>
C	EE	Scholar	An analytic solution to the alibi query in the space-time prisms model for moving object data. (Bart Kuijpers and Rafael Grimson and Walled Othman) <i>International Journal of Geographical Information Science (IJGIS ) [2011] Cited by 3</i>
C	EE	Scholar	A Scaleless Data Model for Direct and Progressive Spatial Query Processing. (Sai Sun and Sham Prasher and Xiaofang Zhou) <i>International Conference on Conceptual Modeling (ER ) [2004] Cited by 2</i>
C	EE	Scholar	Efficient Strip-Mode SAR Raw-Data Simulation of Fixed and Moving Targets. (Ozan Dogan and Mesut Kartal) <i>IEEE Geoscience and Remote Sensing Letters (LGRS ) [2011]</i>
			Computational data modeling for network-constrained moving objects. (I aurvnas Sneicvs and Christian S. Jensen and Augustas Kliavs) <i>GIS</i>

## • 同义词 / 相关词处理

- 中文的同义词 / 相关词处理及相关词典：
  - HowNet (知网) : [http://www.keenage.com/html/c\\_index.html](http://www.keenage.com/html/c_index.html)
    - 包含6万个汉字、1.1万个词语、句法结构式58个.....
  - Chinese WordNet (繁体) : <http://lope.linguistics.ntu.edu.tw/cwn2/>
  - 大词林: <http://www.bigcilin.com/browser/>
- 在实际工作中, 可以根据需要选择中文语义词库。

# 本章小结

## 网页文本处理

- 词条化处理
  - 概念与挑战、常用分词方法、分词工具
- 停用词处理
- 规范化处理
  - 词干提取、拼写检查、相关词处理