信息检索与Web搜索

第4讲 词典及容错式检索

Dictionary and tolerant retrieval

授课人：高曙明

*改编自“现代信息检索”网上公开课件（http://ir.ict.ac.cn/~wangbin）

词典

- 用于存储词项词汇表的数据结构
- 采用定长数组的词典结构

词项	文档频率	指向倒排记录表的指针
a	656 265	→
aachen	65	→
…	…	…
zulu	221	→

空间消耗： 20字节 4字节 4字节

快速词项定位

- 给定查询词项“信息”，如何在词典中快速找到这个词项?
- 需要支持快速查找的词典数据结构
  - 哈希表方式
  - 搜索树方式
- 选择何种方式需要考虑以下因素:
  - 词项数目是否固定或者说词项数目是否持续增长?
  - 词项的相对访问频率如何?
  - 词项的数目有多少?

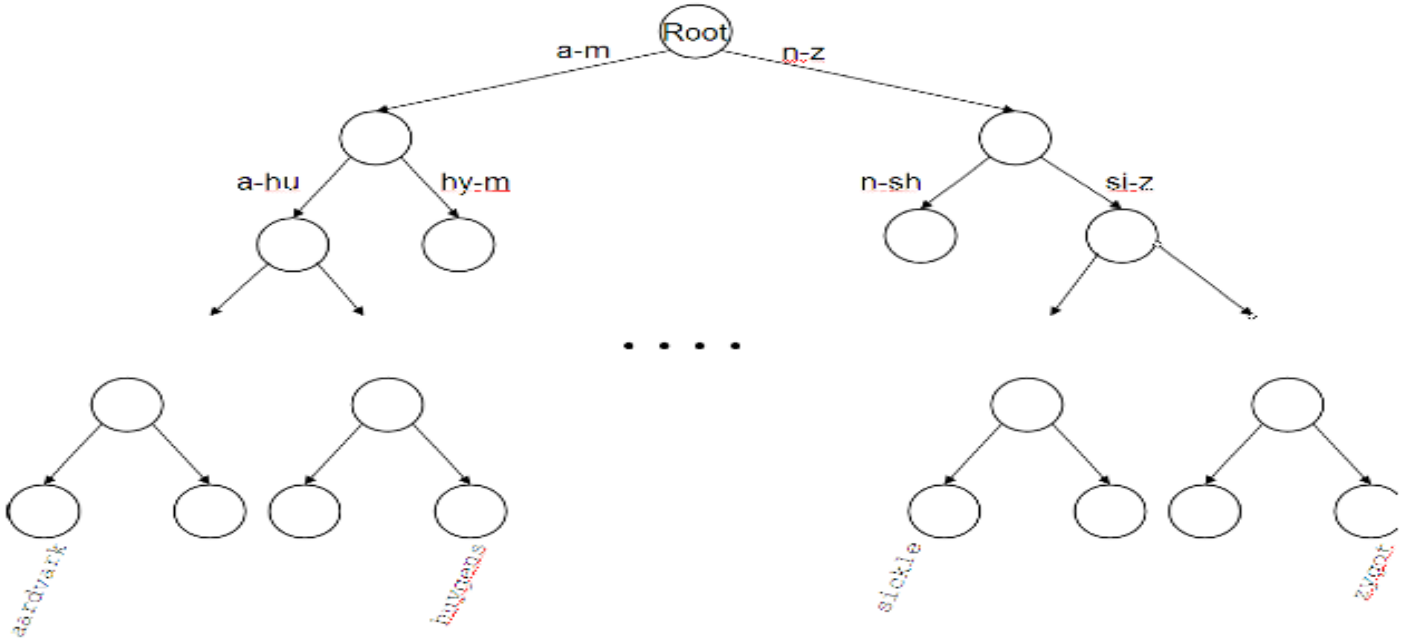
基于哈希表的词典索引

- **方法：**将每个词项通过哈希函数映射成一个整数
- **查询处理：** 对查询词项进行哈希，如果有冲突，则解决冲突，最后在定长数组中定位
- **优点：**在哈希表中的定位速度快于树中的定位速度
  - 查询时间是常数
- **缺点：**
  - 不支持前缀搜索 (比如所有以automat开头的词项)
  - 如果词汇表不断增大，需要定期对所有词项重新哈希

基于搜索树的词典索引

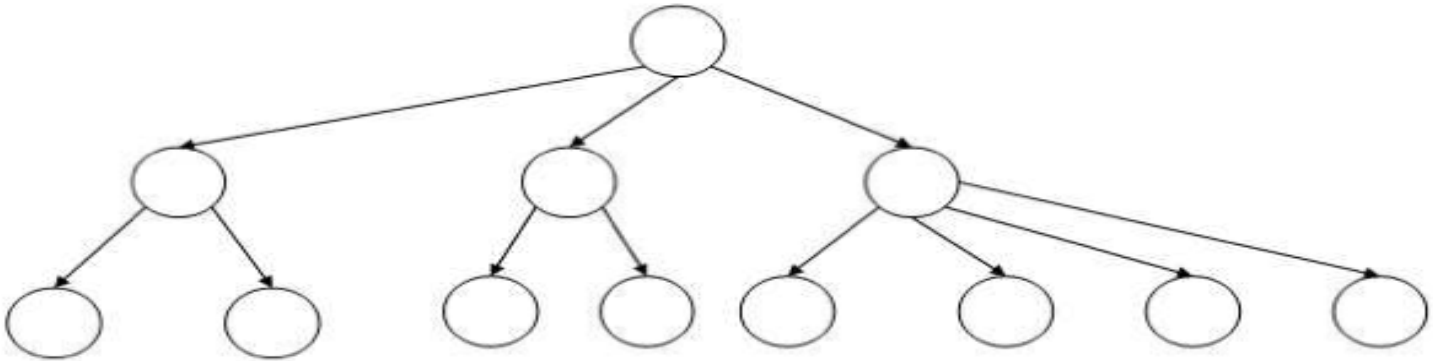
- **方法：** 采用搜索树作为词典的索引，一般采用平衡二叉树
- **优点：** 可以支持前缀查找
- **缺点：** 搜索速度略低于哈希表方式：O(logM), 其中M是词汇数
  使二叉树重新保持平衡开销很大
- **改进：** 采用 B-树减轻上述问题
- **B-树定义：** 每个内部节点的子节点数目在 [a, b]之间，
  其中 a, b 为合适的正整数, e.g., [2, 4]

二叉搜索树示例



平衡性，字符集有预定的排序方式

B-树示例



子节点数目在[2,4]区间

通配查询

- **通配查询：** 指词项中带有通配符“*”的查询
- **分类：**
  - 尾通配符查询 mon* :找出所有包含以 mon开头的词项的文档
  - 首通配符查询 *mon: 找出所有包含以*mon*结尾的词项的文档
  - 中间通配符查询 m*nchen：找出所有包含以m开头并以nchen结尾的词项的文档
- **作用：** 满足用户的以下需求
  - 用户需要进行拼写不确定的查询
  - 用户需要查找某个查询词项的所有变形

通配查询的处理

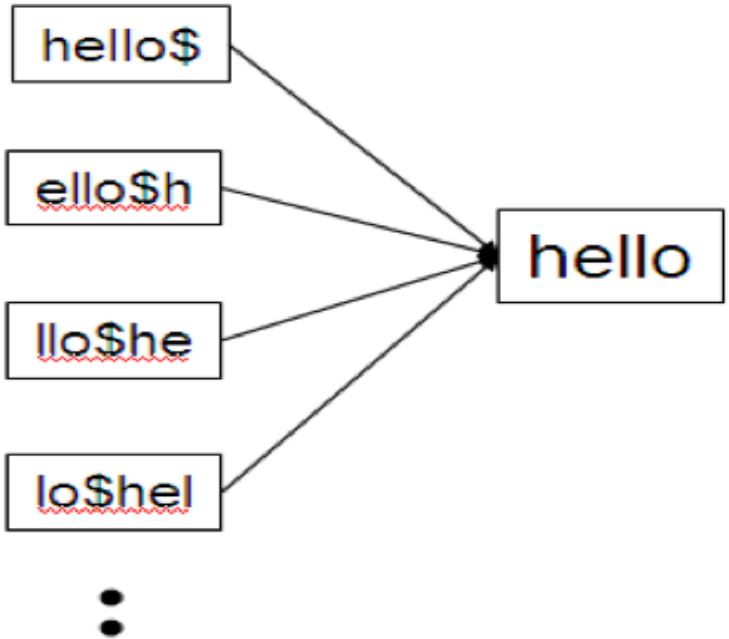
□ 处理方法:

- mon*：采用B-树词典结构，搜索区间mon ≤ t < moo上的所有词项t， 返回相关文档
- *mon：将所有的词项倒转过来，然后基于它们建立一棵辅助的B树； 返回区间nom ≤ t < non上的词项t
- m*nchen： 在B-树和反向B树中分别查找满足m*和 *nchen的词项集合，然后求交集

轮排索引(Permuterm index)

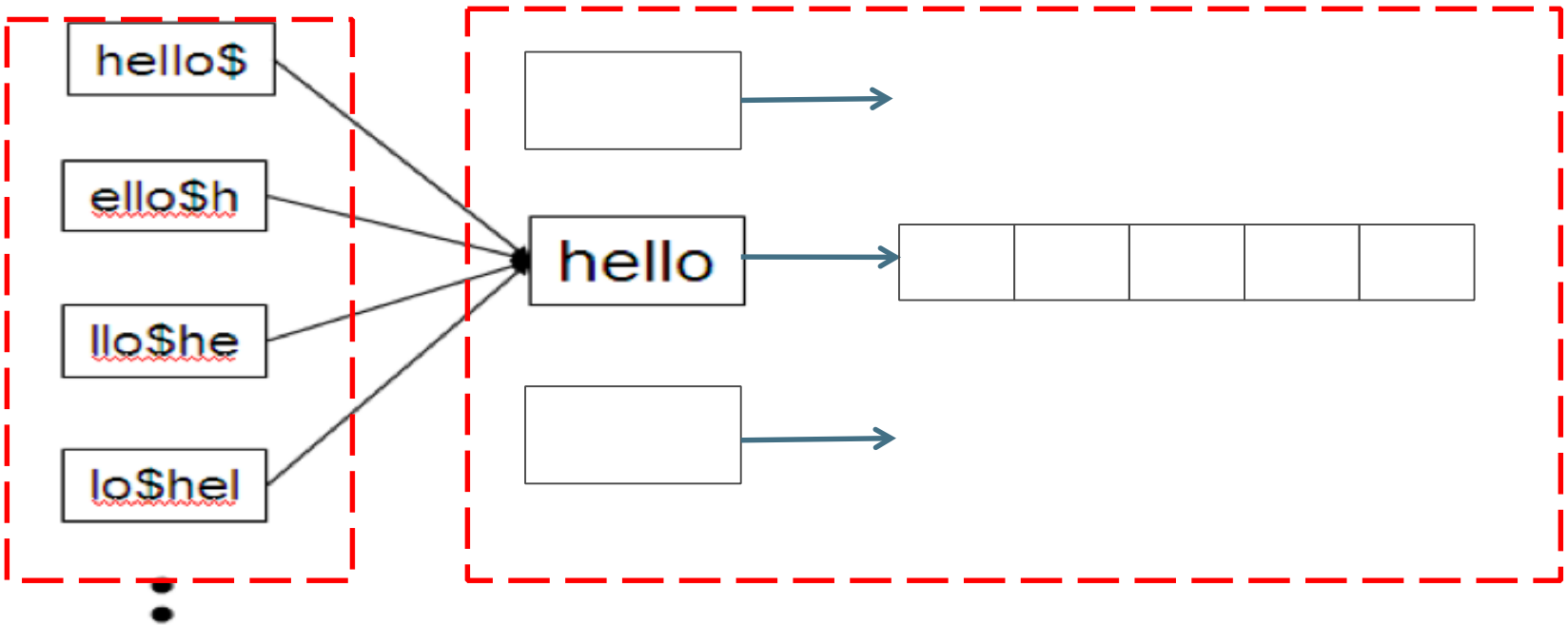
- **轮排词汇集：**将一常规词项旋转后得到的集合
- **举例：**对于词项hello，（hello$, ello$h, llohe, lohel, 和o$hell ）是其轮排词汇集，其中 $ 是一个特殊符号，用于标识词项结束，每个扩展词都指向原始词项
- **轮排索引：**将轮排词汇集加入搜索树形成的词典索引
- **作用：**有效地支持一般性通配符查询

轮排索引(Permuterm index)



- 轮排结果到词项的映射示意图
- 相对于通常的B-树，轮排树的空间要大4倍以上

轮排索引(Permuterm index)



轮排索引（通配查询→词项，采用B树组织）　　传统倒排索引（词项→文档）

基于轮排索引的通配符查询

- **方法：**首先将通配查询词项进行旋转，使*出现在末尾，然后在轮排索引的B-树中搜索，并将搜索结果映射到常规词项
- **举例：**
  - 对于 *X, 查询 X$*
  - 对于 X*Y, 查询 Y$X*
  - 对于hel*o，查询 o$hel*
  - 对于 fi*mo*er，先查er$fi*，再除去不包含mo的词项

k-gram索引

- **K-gram:** 由K个字符组成的序列（即长度为K的字符序列）
- **2-gram:** 由2个字符组成的序列，又称为二元组(bigram)
- **词项的K-grams:** 由词项中所有连读的k个字符构成的k-gram组成的集合
- **例子:** April的bigrams: $a ap pr ri il l$
- $ 是一个特殊字符，标识词项的开始或结束

k-gram索引

- **K-gram索引：** 是一种特殊的倒排索引结构，其中词典由词汇表中所有词项的所有K-gram构成，每个倒排记录表则由包含该K-gram的所有词项组成
- **例子：** 3-gram(trigram)索引

- **本质：** 对词项构建一个倒排索引(二级索引)
- **优点：** 比轮排索引空间开销要小

基于K-gram索引的通配符查询

- **主要步骤：**
  - 对给定的带*的词项，生成其所有的K-gram
  - 基于K-gram索引，找出包含上述所有K-gram的词项集
  - 通过与给定词项进行字符串匹配，对词项集进行过滤
  - 用余下的词项在词项-文档倒排索引中查找文档
- **例子：查询mon***
  - 执行布尔查询: $m AND mo AND on
  - 所有以前缀mon开始的词项被返回，当然也包含许多伪正例，比如MOON，因此，必须要做后续的过滤处理

拼写校正

- **任务目标：** 对用户输入的错误查询词项进行纠正，通过纠正用户的查询，提高检索效果
- **两种方法**
  - 词独立纠错(Isolated word)法
    - 只检查每个单词本身的拼写错误
    - 如果某个单词拼写错误后变成另外一个单词，则无法查出，
      e.g., an asteroid that fell form the sky
  - 上下文敏感(Context-sensitive)法
    - 纠错时考虑周围的单词
    - 能纠正上例中的错误 form/from

词独立纠错法

- **基本思想：**对于一个需要纠错的查询词，在其可能的正确拼写中，选择距离最近中的最常见的一个
  - 可能正确拼写的来源：文档集上的词项词汇表
  - 计算两个词的邻近度（相似度）
  - 最常见的选择：词频（文档集中或用户查询记录中）
- **核心问题：**词之间的邻近度计算
- **两种方法：**
  - 基于编辑距离的邻近度计算
  - 基于k-gram重合度的邻近度计算

基于编辑距离的邻近度计算

- **编辑距离(Edit distance或者Levenshtein distance)：** 两个字符串s_1和s_2之间的编辑距离是指从 s_1 转换成s_2所需要的最少的基本操作数目
- **基本操作：** 插入(insert)、删除(delete)、替换(replace)
- **例子：** cat-cart: 1；cat-cut: 1；cat-act: 2
- **计算方法：** 采用动态规划算法进行计算

Levenshtein距离：实例

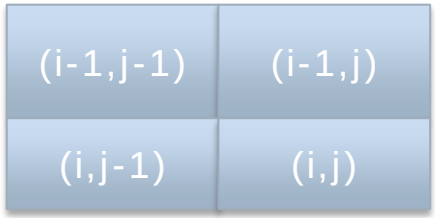
		f	a	s	t
	0	1	2	3	4
c	1	1	2	3	4
a	2	2	1	2	3
t	3	3	2	2	2
s	4	4	3	2	3

cats中的c、t、s分别用f、s、t替换，即可得到fast，所以代价是3，即距离是3

Levenshtein距离：算法

```
LevenshteinDistance(s1, s2)
  1  for i ← 0 to |s1|
  2  do m[i, 0] = i
  3  for j ← 0 to |s2|
  4  do m[0, j] = j
  5  for i ← 1 to |s1|
  6  do for j ← 1 to |s2|
  7       do if s1[i] = s2[j]
  8            then m[i,j] = min{m[i-1,j]+1, m[i,j-1]+1, m[i-1,j-1]}
  9            else  m[i,j] = min{m[i-1,j]+1, m[i,j-1]+1, m[i-1,j-1]+1}
 10  return m[|s1|, |s2|]
```

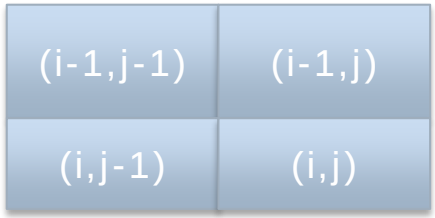
Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)



Levenshtein距离：算法

```
LevenshteinDistance(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7       do if s1[i] = s2[j]
 8            then m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]}
 9            else  m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1}
10  return m[|s1|, |s2|]
```

Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)

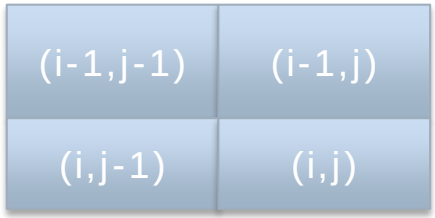


左邻居

Levenshtein距离：算法

```
LevenshteinDistance(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7       do if s1[i] = s2[j]
 8            then m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]}
 9            else  m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1}
10  return m[|s1|, |s2|]
```

Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)



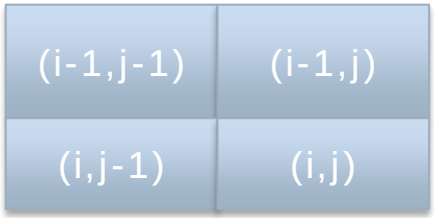
上邻居

Levenshtein距离：算法

LEVENSHTEINDISTANCE(s_1, s_2)

```
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7      do if s1[i] = s2[j]
 8           then m[i,j] = min{m[i-1,j]+1, m[i,j-1]+1, m[i-1,j-1]}
 9           else  m[i,j] = min{m[i-1,j]+1, m[i,j-1]+1, m[i-1,j-1]+1}
10  return m[|s1|, |s2|]
```

Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)



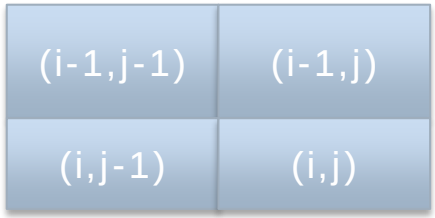
左上邻居

Levenshtein距离：算法

LevenshteinDistance(s_1, s_2)

```
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7      do if s1[i] = s2[j]
 8           then m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]}
 9           else  m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1}
10  return m[|s1|, |s2|]
```

Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)



左上邻居

Levenshtein矩阵单元的组成

从左上角邻居到来的开销 (copy 或 replace)	从上方邻居到来的代价 (delete)
从左方邻居到来的代价 (insert)	上述三者之中最低的代价

Levenshtein 距离：例子

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 / 1				
s	2 / 2				
l	3 / 3				
o	4 / 4				

27

Levenshtein 距离：例子

		s	n	o	w
	0	**1** / **1**	**2** / **2**	**3** / **3**	**4** / **4**
o	**1** / **1**	1 / 2 / 2 / ?			
s	**2** / **2**				
l	**3** / **3**				
o	**4** / **4**				

28

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** / **1**	**1** / 2	2 / **1**						
s	**2** / **2**								
l	**3** / **3**								
o	**4** / **4**								

29

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** / **1**	**1** / 2	2 / **1**	2 / 2	3 / ?				
s	**2** / **2**								
l	**3** / **3**								
o	**4** / **4**								

30

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** / **1**	**1** / 2	2 / **1**	**2** / **2**	3 / **2**				
s	**2** / **2**								
l	**3** / **3**								
o	**4** / **4**								

31

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	2 3	4 ?		
s	**2** **2**								
l	**3** **3**								
o	**4** **4**								

32

Levenshtein 距离：例子

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 / 1	1 2 / 2 1	2 3 / 2 2	2 4 / 3 2	
s	2 / 2				
l	3 / 3				
o	4 / 4				

33

Levenshtein 距离：例子

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / 3 ?
s	**2** **2**				
l	**3** **3**				
o	**4** **4**				

34

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**								
l	**3** **3**								
o	**4** **4**								

35

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	1 3	2 ?						
l	**3** **3**								
o	**4** **4**								

36

Levenshtein 距离：例子

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** **2**	**1** 2 / 3 **1**			
l	**3** **3**				
o	**4** **4**				

37

Levenshtein 距离：例子

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** / **2**	**1** 2 / 3 **1**	2 3 / 2 ?		
l	**3** / **3**				
o	**4** / **4**				

38

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** / **1**	**1** / 2	2 / **1**	**2** / **2**	3 / **2**	**2** / 3	4 / **2**	4 / **3**	5 / **3**
s	**2** / **2**	**1** / 3	2 / **1**	**2** / **2**	3 / **2**				
l	**3** / **3**								
o	**4** / **4**								

39

Levenshtein 距离：例子

		s	n	o	w
	0	**1** 1	**2** 2	**3** 3	**4** 4
o	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** / **2**	**1** 2 / 3 **1**	**2** 3 / **2** **2**	3 3 / 3 ?	
l	**3** / **3**				
o	**4** / **4**				

40

Levenshtein 距离：例子

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** **2**	**1** 2 / 3 **1**	**2** 3 / **2** **2**	**3** **3** / **3** **3**	
l	**3** **3**				
o	**4** **4**				

41

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	3 4	4 ?
l	**3** **3**								
o	**4** **4**								

42

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1** / **1**	**1** / 2	2 / **1**	**2** / **2**	3 / **2**	**2** / 3	4 / **2**	4 / **3**	5 / **3**
s		**2** / **2**	**1** / 2	3 / **1**	**2** / **2**	3 / **2**	**3** / **3**	4 / **3**	**3** / 4	4 / **3**
l		**3** / **3**								
o		**4** / **4**								

43

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1** / **1**	**1** / 2	2 / **1**	**2** / **2**	3 / **2**	**2** / 3	4 / **2**	4 / **3**	5 / **3**
s		**2** / **2**	**1** / 3	2 / **1**	**2** / **2**	3 / **2**	**3** / **3**	3 / **3**	**3** / 4	4 / **3**
l		**3** / **3**	3 / 4	2 / ?						
o		**4** / **4**								

44

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l	**3** **3**	3 4	**2** **2**						
o	**4** **4**								

45

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1** / **1**	**1** / 2	2 / **1**	**2** / **2**	3 / **2**	**2** / 3	4 / **2**	4 / **3**	5 / **3**
s		**2** / **2**	**1** / 3	2 / **1**	**2** / **2**	3 / **2**	**3** / **3**	**3** / **3**	**3** / 4	4 / **3**
l		**3** / **3**	3 / 4	**2** / **2**	2 / 3	3 / ?				
o		**4** / **4**								

46

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l	**3** **3**	3 4	**2** **2**	**2** 3	3 **2**				
o	**4** **4**								

47

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1**	**1**	2	**2**	3	**2**	4	4	5
	1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s	**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
	2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l	**3**	3	**2**	**2**	3	3	4		
	3	4	**2**	3	**2**	3	?		
o	**4**								
	4								

48

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1** / **1**	**1** / 2	2 / **1**	**2** / **2**	3 / **2**	**2** / 3	4 / **2**	4 / **3**	5 / **3**
s		**2** / **2**	**1** / 2	3 / **1**	**2** / **2**	3 / **2**	**3** / **3**	4 / **3**	**3** / 4	4 / **3**
l		**3** / **3**	3 / **2**	4 / **2**	**2** / 3	3 / **2**	**3** / 4	**3** / **3**		
o		**4** / **4**								

49

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** / **1**	**1** / 2	**1**	**2** / **2**	3 / **2**	**2** / 3	4 / **2**	4 / **3**	5 / **3**
s	**2** / **2**	**1** / 3	2 / **1**	**2** / **2**	3 / **2**	**3** / **3**	**3** / **3**	**3** / 4	4 / **3**
l	**3** / **3**	3 / 4	**2** / **2**	**2** / 3	3 / **2**	**3** / **3**	4 / **3**	4 / 4	4 / ?
o	**4** / **4**								

50

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1**	**1**	2	**2**	3	**2**	4	4	5
		1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s		**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
		2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l		**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
		3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o		**4**								
		4								

51

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l	**3** **3**	3 4	**2** **2**	**2** 3	3 **2**	**3** **3**	4 **3**	**4** **4**	**4** **4**
o	**4** **4**	4 5	3 ?						

52

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l	**3** **3**	3 4	**2** **2**	**2** 3	3 **2**	**3** **3**	4 **3**	**4** **4**	**4** **4**
o	**4** **4**	4 5	**3** **3**						

53

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l	**3** **3**	3 4	**2** **2**	**2** 3	3 **2**	**3** **3**	4 **3**	**4** **4**	**4** **4**
o	**4** **4**	4 5	**3** **3**	3 4	3 ?				

54

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s	**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l	**3** **3**	3 4	**2** **2**	**2** 3	3 **2**	**3** **3**	4 **3**	**4** **4**	**4** **4**
o	**4** **4**	4 5	**3** **3**	**3** 4	**3** **3**				

55

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1**	**1**	2	**2**	3	**2**	4	4	5
		1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s		**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
		2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l		**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
		3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o		**4**	4	**3**	**3**	**3**	2	4		
		4	5	**3**	4	**3**	4	?		

56

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1**	**1**	2	**2**	3	**2**	4	4	5
	1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s	**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
	2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l	**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
	3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o	**4**	4	**3**	**3**	**3**	**2**	4		
	4	5	**3**	4	**3**	4	**2**		

57

Levenshtein 距离：例子

		s		n		o		w	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o	**1**	**1**	2	**2**	3	**2**	4	4	5
	1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s	**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
	2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l	**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
	3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o	**4**	4	**3**	**3**	**3**	**2**	4	4	5
	4	5	**3**	4	**3**	4	**2**	3	?

58

Levenshtein 距离：例子

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1**	**1**	2	**2**	3	**2**	4	4	5
		1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s		**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
		2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l		**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
		3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o		**4**	4	**3**	**3**	**3**	**2**	4	4	5
		4	5	**3**	4	**3**	4	**2**	**3**	**3**

59

Levenshtein 距离：例子

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** / **2**	**1** 2 / 3 **1**	**2** 3 / **2** **2**	**3** **3** / **3** **3**	**3** 4 / 4 **3**
l	**3** / **3**	3 **2** / 4 **2**	**2** 3 / 3 **2**	**3** 4 / **3** **3**	**4** **4** / **4** **4**
o	**4** / **4**	4 **3** / 5 **3**	**3** **3** / 4 **3**	**2** 4 / 4 **2**	4 5 / 3 **3**

60

Levenshtein距离：编辑路径举例

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1**	**1**	2	**2**	3	**2**	4	4	5
		1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s		**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
		2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l		**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
		3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o		**4**	4	**3**	**3**	**3**	**2**	4	4	5
		4	5	**3**	4	**3**	4	**2**	**3**	**3**

cost	operation	input	output
1	insert	*	w

61

Levenshtein距离：编辑路径举例

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1** **1**	**1** 2	2 **1**	**2** **2**	3 **2**	**2** 3	4 **2**	4 **3**	5 **3**
s		**2** **2**	**1** 3	2 **1**	**2** **2**	3 **2**	**3** **3**	**3** **3**	**3** 4	4 **3**
l		**3** **3**	3 4	**2** **2**	**2** 3	3 **2**	**3** **3**	4 **3**	**4** **4**	**4** **4**
o		**4** **4**	4 5	**3** **3**	**3** 4	**3** **3**	2 4	4 2	4 3	5 3

cost	operation	input	output
0	(copy)	o	o
1	insert	*	w

62

Levenshtein距离：编辑路径举例

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** / **2**	**1** 2 / 3 **1**	**2** 3 / **2** **2**	**3** **3** / **3** **3**	**3** 4 / 4 **3**
l	**3** / **3**	3 **2** / 4 **2**	**2** 3 / 3 **2**	**3** 4 / **3** **3**	**4** **4** / **4** **4**
o	**4** / **4**	4 **3** / 5 **3**	**3** **3** / 4 **3**	**2** 4 / 4 **2**	4 5 / **3** **3**

cost	operation	input	output
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

63

Levenshtein 距离：编辑路径举例

			s		n		o		w	
		0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
o		**1**	**1**	2	**2**	3	**2**	4	4	5
		1	2	**1**	**2**	**2**	3	**2**	**3**	**3**
s		**2**	**1**	2	**2**	3	**3**	**3**	**3**	4
		2	3	**1**	**2**	**2**	**3**	**3**	4	**3**
l		**3**	3	**2**	**2**	3	**3**	4	**4**	**4**
		3	4	**2**	3	**2**	**3**	**3**	**4**	**4**
o		**4**	4	**3**	**3**	**3**	**2**	4	4	5
		4	5	**3**	4	**3**	4	**2**	**3**	**3**

cost	operation	input	output
0	(copy)	s	s
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

64

Levenshtein距离：编辑路径举例

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** / **2**	**1** 2 / 3 **1**	**2** 3 / **2** **2**	**3** **3** / **3** **3**	**3** 4 / 4 **3**
l	**3** / **3**	3 **2** / 4 **2**	**2** 3 / 3 **2**	**3** 4 / **3** **3**	**4** **4** / **4** **4**
o	**4** / **4**	4 **3** / 5 **3**	**3** **3** / 4 **3**	**2** 4 / 4 **2**	4 5 / **3** **3**

cost	operation	input	output
1	delete	o	*
0	(copy)	s	s
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

65

带权重的编辑距离

- **特点：**对不同字符进行的操作赋予不同的权重
- **必要性分析：**将字符m替换成n与将字符m替换成q应该有区别，前者的权重应该更小

- **目的：**通过考虑出错操作发生概率的因素，提高距离计算准确性

利用编辑距离进行拼写校正

- 给定查询词，穷举词汇表中和该查询的编辑距离(或带权重的编辑距离)低于某个预定值的所有词项
- 将结果推荐给用户
- 代价很大，实际当中往往通过启发式策略提高效率
  - 限制在与查询词具有相同首字母的词项
  - 保证两者之间具有较长公共子串

基于k-gram重合度的邻近度计算

- **k-gram重合度：** 指 | A∩B | / | A∪B | ，其中A、B分别是两个词的*k*-gram集合
- **例子：** bord 与 boardroom之间的2-gram重合度

A=5, B=10

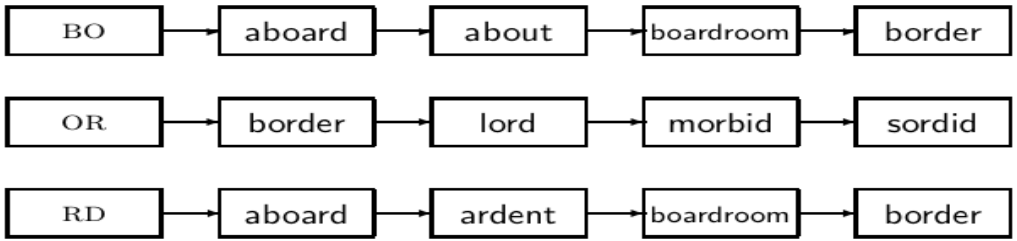
A∩B=3, A∪B=10+5-3=12

重合度 = 3/12

基于K-gram索引的拼写校正

主要步骤

- 确定查询词项的k-gram集合 A
- 利用k-gram索引返回A相关倒排记录表中的词项
- 对去重后的词，计算其与查询词的k-gram重合度
- 根据给定阈值，确定匹配上的正确词项返回



查询词bord的2-gram索引片段

上下文敏感的拼写校正

- 对于flew form Heathrow，如何纠错form?
- **一种方法:** 基于命中数(hit-based)的拼写校正
  - 对于每个查询词项返回 相近的“正确” 词项
  - 组合所有可能，分别进行查询，取具有最高命中数者
  - 搜索 “*fled form* Heathrow”
  - 搜索 “*flew fore* Heathrow”
  - 搜索 “*flew from* Heathrow” 会有最高的结果命中数
- **问题:** 假定 flew有7个可能的候选词, form 有20个， Heathrow 有3个，那么需要穷举出多少个查询?
- **更高效的方法:** 基于查询库确定合理的组合

基于发音的校正（Soundex）

- **目标:** 寻找发音相似的单词，对查询进行扩展，提高检索效果
- 比如，对查询词 chebyshev，将其扩展到 tchebyscheff
- **算法步骤:**
  - 将词典中每个词项转换成一个4字符缩减形式(即进行Soundex编码)，构建词典的soundex索引
  - 对查询词项做同样的处理
  - 基于soundex索引搜索音似词

Soundex 编码算法

1. 保留词项的首字母
2. 将后续所有的A、E、I、O、U、H、W及Y等字母转换为0。
3. 按照如下方式将字母转换成数字：
   - B, F, P, V → 1
   - C, G, J, K, Q, S, X, Z → 2
   - D,T → 3; L → 4; M, N → 5; R → 6
4. 将连续出现的两个同一字符转换为一个字符直至再没有这种现象出现。
5. 在结果字符串中剔除0，并在结果字符串尾部补足0，然后返回前四个字符，该字符由1个字母加上3个数字组成。

例子：HERMAN的Soundex编码

- 保留 H
- *ERMAN* → *0RM0N*
- *0RM0N* → *06505*
- *06505* → *655*
- 返回 *H655*
- **注意: *HERMANN* 会产生同样的编码**

Soundex的应用情况

- 在IR中并不非常普遍
- 适用于“高召回率”任务 (e.g., 国际刑警组织Interpol在全球范围内追查罪犯)
- Zobel and Dart (1996)提出了一个更好的发音匹配方法

参考资料

- 《信息检索导论》第3章、MG4.2
- 高效拼写校正方法:
- K. Kukich. Techniques for automatically correcting words in text. ACM Computing Surveys 24(4), Dec 1992.
- J. Zobel and P. Dart. Finding approximate matches in large lexicons. Software - practice and experience 25(3), March 1995. http://citeseer.ist.psu.edu/zobel95finding.html
- Mikael Tillenius: Efficient Generation and Ranking of Spelling Error Corrections. Master's thesis at Sweden's Royal Institute of Technology. http://citeseer.ist.psu.edu/179155.html

参考资料

- Peter Norvig: How to write a spelling corrector
- http://norvig.com/spell-correct.html
- http://ifnlp.org/ir
- Soundex演示
- Levenshtein距离的演示
- Peter Norvig的拼写校正工具

课后作业

- 见课程网页:

  http://10.76.3.31