自然语言处理

Natural Language Processing

权小军 教授

中山大学数据科学与计算机学院

quanxj3@mail.sysu.edu.cn

应用展示

http://ictclas.nlpir.org/nlpir/

内容

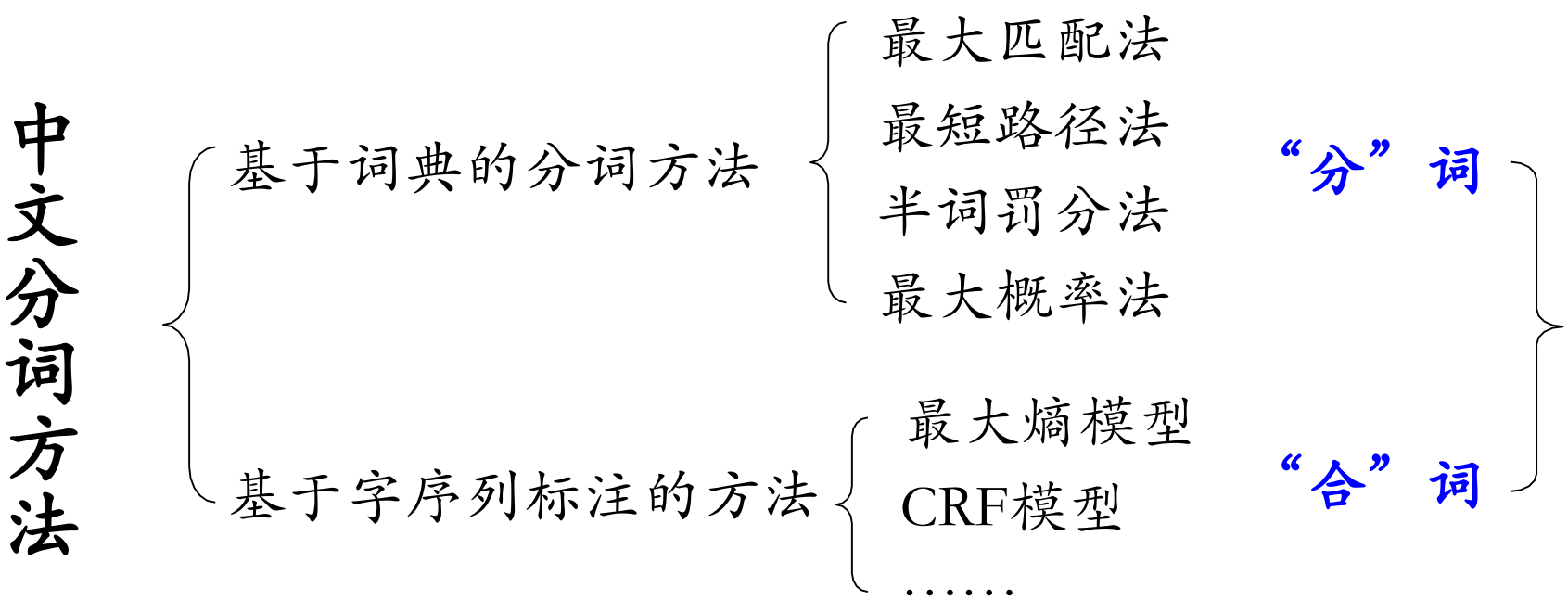
- 中文分词基本方法概述
- 中文分词技术的评测
- 小结

4.1中文分词基本方法概述

中文分词基本方法概述

- 中文分词方法
  - 基于词典的分词方法
    - 最大匹配法
    - 最短路径法
    - 半词罚分法
    - 最大概率法
  - 基于字序列标注的方法
    - 最大熵模型
    - CRF模型
    - ……

中文分词基本方法概述


中文分词方法
基于词典的分词方法
最大匹配法
最短路径法
半词罚分法
最大概率法
“分”词
基于字序列标注的方法
最大熵模型
CRF模型
……
“合”词

4.2 基于词典的分词方法

基于词典的分词方法

1. 基于词典的分词方法

思考：你会怎么做？

基于词典的分词方法

1. 基于词典的分词方法

   a. 最大匹配法

最大匹配法分词示例

输入：S1=“计算语言学课程是两个课时"

输出：S2=" "

词 典
…
计算语言学
课程
课时
…

最大匹配法分词示例

输入：S1=“计算语言学课程是两个课时"

输出：S2= " "

设定最大词长MaxLen = 5

W= 计算语言学

词 典
…
计算语言学
课程
课时
…

最大匹配法分词示例

输入：S1=“计算语言学课程是两个课时"

输出：S2= " "

设定最大词长MaxLen = 5

W= 计算语言学

……

词 典
…
计算语言学
课程
课时
…

最大匹配法分词示例

输入：S1=“计算语言学课程是两个课时"

输出：S2= " "

设定最大词长MaxLen = 5

W1= 计算语言学

……

词 典
…
计算语言学
课程
课时
…

大规模真实语料中99%的词例（token）的长度都在5字以内 [1]

[1] 黄昌宁、赵海，2007，中文分词十年回顾，《中文信息学报》2007年第3期，8-19页。

最大匹配法的问题

- 无法发现分词歧义 → 单向最大匹配改为双向

正向最大匹配和逆向最大匹配结果不同，意味着存在分词歧义。

最大匹配法的问题

- 无法发现分词歧义 → 单向最大匹配改为双向

正向最大匹配和逆向最大匹配结果不同，意味着存在分词歧义。

FMM	有意/	见/	分歧/
BMM	有/	意见/	分歧/

最大匹配法分词的问题

- 双向最大匹配法可以发现链长为奇数的交集型歧义，但无法发现链长为偶数的交集型歧义
  - 正向最大匹配和逆向最大匹配结果相同

FMM & BMM　　原子/　结合/　成分/　子时/

最大匹配法分词的问题

- 双向最大匹配法可以发现链长为奇数的交集型 歧义，但无法发现链长为偶数的交集型歧义
  - 正向最大匹配和逆向最大匹配结果相同

    FMM & BMM　　原子/　结合/　成分/　子时/
- 无法发现组合型歧义

最大匹配法分词的问题

- 双向最大匹配法可以发现链长为奇数的交集型 歧义，但无法发现链长为偶数的交集型歧义
  - 正向最大匹配和逆向最大匹配结果相同

    FMM & BMM　　原子/　结合/　成分/　子时/
- 无法发现组合型歧义
- 在最大匹配法的基础上进行修改，如何给出“改错”的触发条件带有一定的主观性

最大匹配法分词的问题

- 双向最大匹配法可以发现链长为奇数的交集型 歧义，但无法发现链长为偶数的交集型歧义
  - 正向最大匹配和逆向最大匹配结果相同

    FMM & BMM　　原子/　结合/　成分/　子时/
- 无法发现组合型歧义
- 在最大匹配法的基础上进行修改，如何给出“改错”的触发条件带有一定的主观性

需要更全面地考虑分词的改进办法

基于词典的分词方法

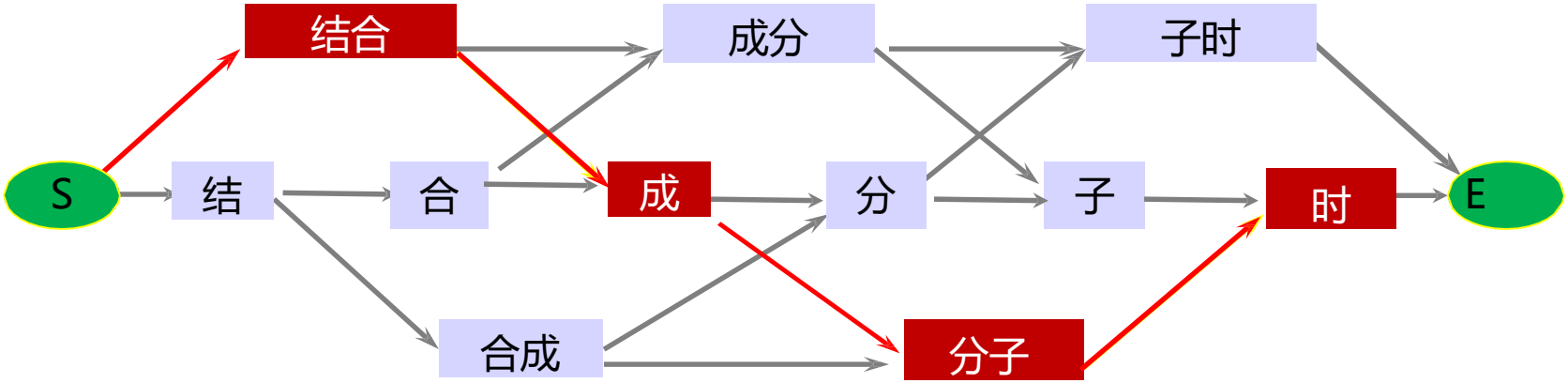
1. 基于词典的分词方法

   a. 最大匹配法

   b. 最优路径法

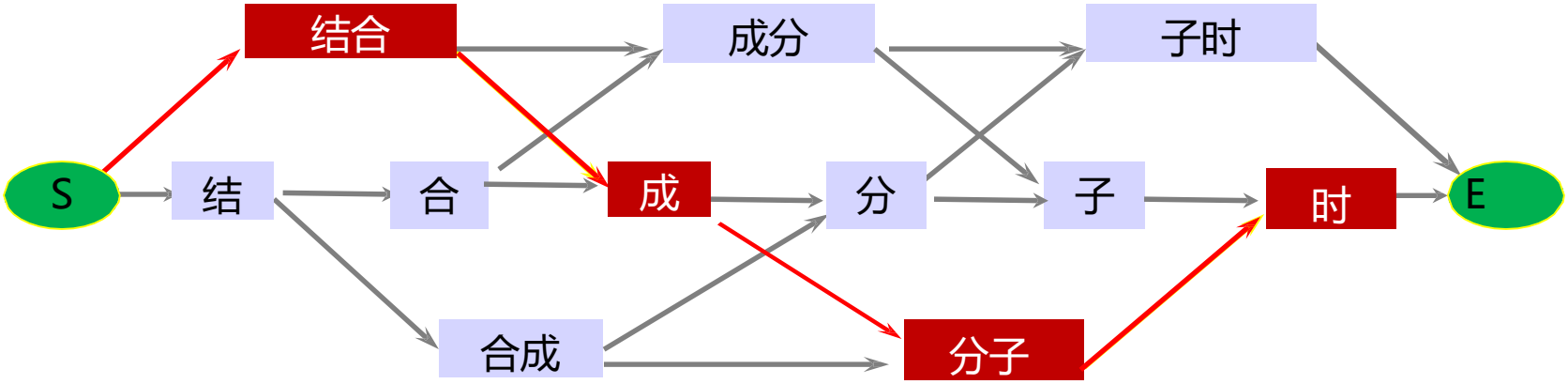
最优路径法

- 看待汉语词语切分问题的新视角：词图上的最优路径求解问题



最优路径法

- 看待汉语词语切分问题的新视角：词图上的最优路径求解问题



- 词图给出了一个字符串的全部切分可能性
- 分词任务：寻找一条起点S到终点E的最优路径

基于词典的分词方法

1. 基于词典的分词方法

a. 最大匹配法

b. 最优路径法

1）词数最少的路径最优

最短路径分词法：词数最少的路径最优

- 基本思想：在词图上选择一条词数最少的路径

最短路径分词法：词数最少的路径最优

- 基本思想：在词图上选择一条词数最少的路径
- 优点：好于单向的最大匹配方法
  - 最大匹配：独立自主|和平|等|互利|的|原则 (6 words)
  - 最短路径：独立自主|和|平等互利|的|原则 (5 words)

最短路径分词法：词数最少的路径最优

- 基本思想：在词图上选择一条词数最少的路径
- 优点：好于单向的最大匹配方法
  - 最大匹配：独立自主|和平|等|互利|的|原则 (6 words)
  - 最短路径：独立自主|和|平等互利|的|原则 (5 words)
- 缺点：同样无法解决大部分交集型歧义

最短路径分词法：词数最少的路径最优

- 基本思想：在词图上选择一条词数最少的路径
- 优点：好于单向的最大匹配方法
  - 最大匹配：独立自主|和平|等|互利|的|原则 (6 words)
  - 最短路径：独立自主|和|平等互利|的|原则 (5 words)
- 缺点：同样无法解决大部分交集型歧义
  - 他说的确实在理

    分词一：？？？

    分词二：？？？

    分词三：？？？

最短路径分词法：词数最少的路径最优

- 基本思想：在词图上选择一条词数最少的路径
- 优点：好于单向的最大匹配方法
  - 最大匹配：独立自主|和平|等|互利|的|原则 (6 words)
  - 最短路径：独立自主|和|平等互利|的|原则 (5 words)
- 缺点：同样无法解决大部分交集型歧义
  - 他说的确实在理

    分词一：他|说|的|确实|在理

    分词二：他|说|的确|实在|理

    分词三：他|说|的确|实|在理

基于词典的分词方法

1. 基于词典的分词方法

a. 最大匹配法

b. 最优路径法

1）词数最少的路径最优

2）半词法

半词法分词：词数最少且半词最少

大多数单字在语境里如果能组成合适的词就不倾向于单独使用！

基本概念	半词	如果一个字不单独作为词使用，就是半词。
	整词	如果一个字更倾向于自己成词而不倾向于和别的字组成词，这类“单字词”就称之为“整词”。这类词就是一般说的单字高频成词语素，比如“人、说、我”等。
基本思路	充分利用半词和整词的差别，尽量选择没有半词落单的分词方案。	

半词法分词

- 在词图的路径优劣评判中引入罚分机制

半词法分词

- 在词图的路径优劣评判中引入罚分机制
- 罚分规则：

1) 每个词对应的边罚1分。

半词法分词

- 在词图的路径优劣评判中引入罚分机制
- 罚分规则：

1) 每个词对应的边罚1分。
2) 每个半词对应的边加罚1分。

半词法分词

- 在词图的路径优劣评判中引入罚分机制
- 罚分规则：

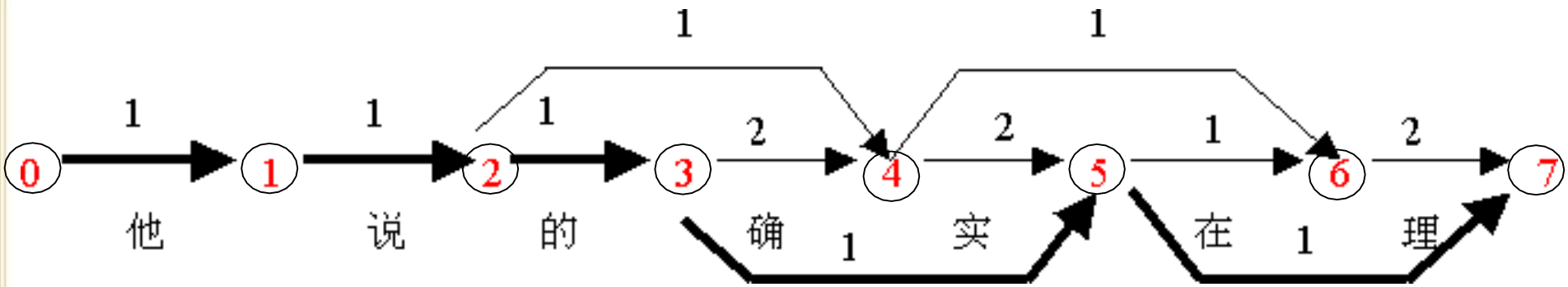
1) 每个词对应的边罚1分。
2) 每个半词对应的边加罚1分。
3) 一个分词方案的评分为它所对应的路径上所有边的罚分之和。

半词法分词

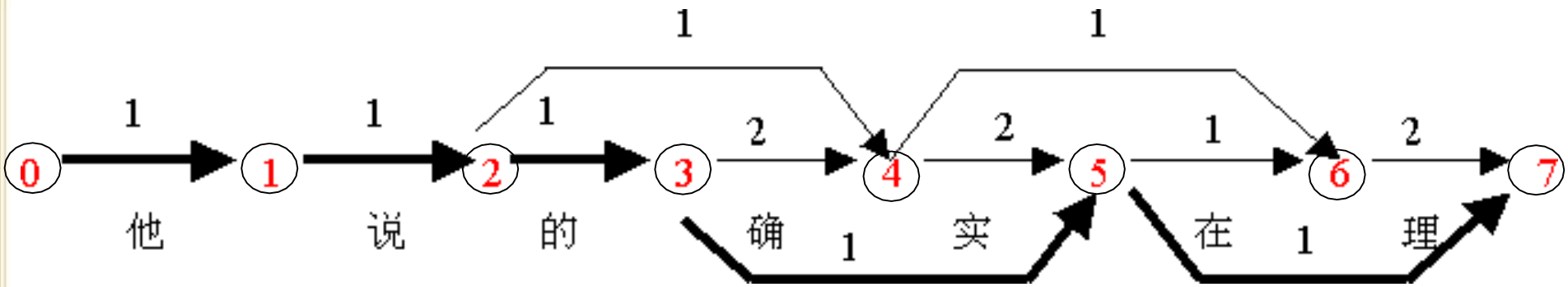
- 在词图的路径优劣评判中引入罚分机制
- 罚分规则：

1) 每个词对应的边罚1分。
2) 每个半词对应的边加罚1分。
3) 一个分词方案的评分为它所对应的路径上所有边的罚分之和。
4) 最优路径就是罚分最低的分词路径。

半词法分词



半词法分词

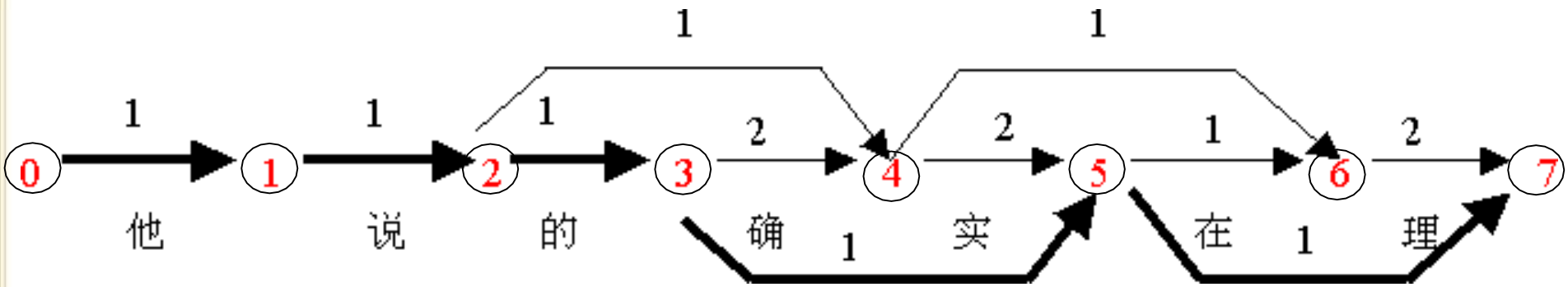


他|说|的|确实|在理 （1+1+1+1+1＝5分）

他|说|的确|实|在理 （1+1+1+2+1＝6分）

他|说|的确|实在|理 （1+1+1+1+2＝6分）

半词法分词



他|说|的|确实|在理 （1+1+1+1+1＝5分）

他|说|的确|实|在理 （1+1+1+2+1＝6分）

他|说|的确|实在|理 （1+1+1+1+2＝6分）

但是：仍然无法解决“有意见分歧”的问题!

基于词典的分词方法

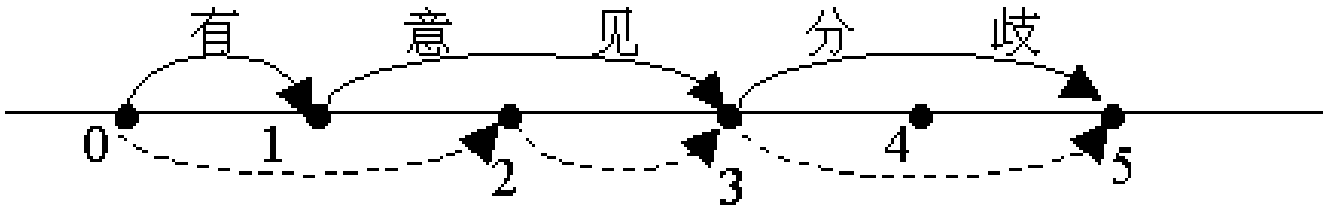
1. 基于词典的分词方法

a. 最大匹配法

b. 最优路径法
   - 1）词数最少的路径最优
   - 2）半词法
   - 3）最大概率法分词

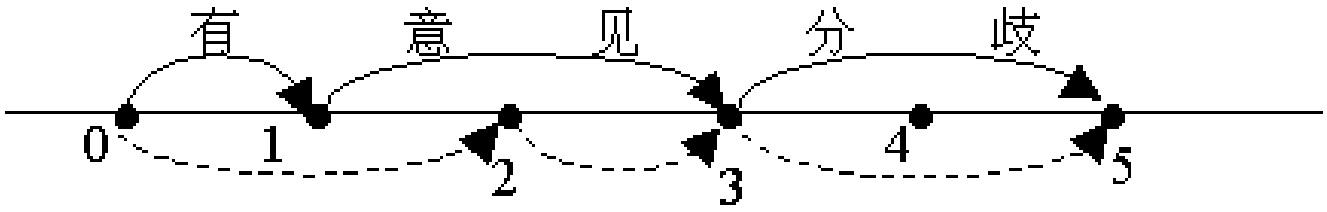
最大概率法分词：字串成词概率最大的路径最优

基本思想：在词图上选择词串概率最大的分词路经作为最优结果



最大概率法分词：字串成词概率最大的路径最优

基本思想：在词图上选择词串概率最大的分词路经作为最优结果



路径1：0－1－3－5

路径2：0－2－3－5

该走哪条路呢?

最大概率法分词

输入字符串S: 有意见分歧

词串W1: 有/意见/分歧/

词串W2: 有意/见/分歧/

输出：？？？

最大概率法分词

输入字符串S: 有意见分歧

词串W1: 有/意见/分歧/

词串W2: 有意/见/分歧/

输出：？？？ **Max(P(W1|S), P(W2|S))?**

最大概率法分词

输入字符串S: 有意见分歧

词串W1: 有/意见/分歧/

词串W2: 有意/见/分歧/

输出：？？？ **Max(P(W1|S), P(W2|S))?**

$$P(W|S)=\frac{P(S|W)\times P(W)}{P(S)}\approx P(W)$$

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

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

最大概率法分词

输入字符串S: 有意见分歧

词串W1: 有/意见/分歧/

词串W2: 有意/见/分歧/

输出：？？？ **Max(P(W1|S), P(W2|S))?**

$$P(W \mid S) = \frac{P(S \mid W) \times P(W)}{P(S)} \approx P(W)$$

独立性假设

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

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

最大概率法分词示例

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

最大概率法分词示例

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

$$P(W1) = P(有) * P(意见) * P(分歧)$$

$$= 1.8 \times 10^{-9}$$

$$P(W2) = P(有意) * P(见) * P(分歧)$$

$$= 1.0 \times 10^{-11}$$

P(W1) > P(W2)

最大概率法分词示例

问题：怎么找出概率最大的分词序列?

用动态规划算法求解最优路径

- 动态规划算法：最优路径中的第 i 个词 W_i 的累积概率等于它的左邻词W_{i-1}的累积概率乘以W_i自身的概率。

$$P'(w_i) = P'(w_{i-1}) \times P(w_i)$$

用动态规划算法求解最优路径

- 动态规划算法：最优路径中的第 i 个词 W_i 的累积概率等于它的左邻词 W_{i-1} 的累积概率乘以 W_i 自身的概率。

$$P'(w_i) = P'(w_{i-1}) \times P(w_i)$$

- 为方便计算，一般把概率转化为路径代价

$$C = -\log(P)$$

$$C'(w_i) = C'(w_{i-1}) + C(w_i)$$

公式1

$C'(w_i)$：最小累积代价　$C'(w_{i-1})$：最佳左邻词

最大概率法算法流程

1) 对一个待分词的字串 S，按照从左到右的顺序取出全部候选词$w_1, w_2, \cdots, w_i, \cdots, w_n$；

最大概率法算法流程

1) 对一个待分词的字串 S，按照从左到右的顺序取出全部候选词$w_1, w_2, \cdots, w_i, \cdots, w_n$；

2) 到词典中查出每个候选词的概率值$P(w_i)$，转换为代价$C(w_i)$，并记录每个候选词的全部左邻词；

最大概率法算法流程

1) 对一个待分词的字串S，按照从左到右的顺序取出全部候选词$w_1, w_2, \cdots, w_i, \cdots, w_n$；
2) 到词典中查出每个候选词的概率值$P(w_i)$，转换为代价$C(w_i)$，并记录每个候选词的全部左邻词；
3) 按照公式1计算每个候选词的累计代价，同时比较得到每个候选词的最佳左邻词；

最大概率法算法流程

1) 对一个待分词的字串 S，按照从左到右的顺序取出全部候选词$w_1, w_2, \cdots, w_i, \cdots, w_n$；
2) 到词典中查出每个候选词的概率值$P(w_i)$，转换为代价$C(w_i)$，并记录每个候选词的全部左邻词；
3) 按照公式1计算每个候选词的累计代价，同时比较得 到每个候选词的最佳左邻词；
4) 如果当前词w_n是字串S的尾词，且累计代价$C'(w_n)$最小，则w_n 就是S的终点词；

最大概率法算法流程

1) 对一个待分词的字串 S，按照从左到右的顺序取出全部候选词$w_1, w_2, \cdots, w_i, \cdots, w_n$；
2) 到词典中查出每个候选词的概率值$P(w_i)$，转换为代价$C(w_i)$，并记录每个候选词的全部左邻词；
3) 按照公式1计算每个候选词的累计代价，同时比较得 到每个候选词的最佳左邻词；
4) 如果当前词w_n是字串S的尾词，且累计代价$C'(w_n)$最小，则w_n 就是S的终点词；
5) 从w_n开始，按照从右到左顺序，依次将每个词的最佳左邻词输出，即为S的分词结果。

最大概率法分词示例

输入字符串S：结合成分子时

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0
3	合成	4.194	7.767	0

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0
3	合成	4.194	7.767	0
4	成	2.800	6.343	1

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0
3	合成	4.194	7.767	0
4	成	2.800	6.343	1
5	成分	3.908	7.451	1
6	分	2.862	9.205	4
7	分子	3.465	9.808	4

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0
3	合成	4.194	7.767	0
4	成	2.800	6.343	1
5	成分	3.908	7.451	1
6	分	2.862	9.205	4
7	分子	3.465	9.808	4
8	子	3.304	10.755	5
9	子时	6.000	13.451	5

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0
3	合成	4.194	7.767	0
4	成	2.800	6.343	1
5	成分	3.908	7.451	1
6	分	2.862	9.205	4
7	分子	3.465	9.808	4
8	子	3.304	10.755	5
9	子时	6.000	13.451	5
10	时	2.478	12.286	7

最大概率法分词示例

输入字符串S：结合成分子时

序号	候选词	代价	累计代价	最佳左邻
0	结	3.573	3.573	-1
1	结合	3.543	3.543	-1
2	合	3.518	7.091	0
3	合成	4.194	7.767	0
4	成	2.800	6.343	1
5	成分	3.908	7.451	1
6	分	2.862	9.205	4
7	分子	3.465	9.808	4
8	子	3.304	10.755	5
9	子时	6.000	13.451	5
10	时	2.478	12.286	7

最大概率法分词的问题

- 并不能解决所有的交集型歧义问题

  例：这事的确定不下来

最大概率法分词的问题

- 并不能解决所有的交集型歧义问题

例：这事的确定不下来

W1= 这/ 事/ 的确/ 定/ 不/ 下来/

W2= 这/ 事/ 的/ 确定/不/下来/

P(W1) < P(W2)

最大概率法分词的问题

- 并不能解决所有的交集型歧义问题

  例：这事的确定不下来

  W1= 这/ 事/ 的确/ 定/ 不/ 下来/

  W2= 这/ 事/ 的/ 确定/不/下来/

  P(W1) < P(W2)

- 一般也无法解决组合型歧义问题

最大概率法分词的问题

- 并不能解决所有的交集型歧义问题

  例：这事的确定不下来

  W1= 这/ 事/ 的确/ 定/ 不/ 下来/

  W2= 这/ 事/ 的/ 确定/不/下来/

  P(W1) < P(W2)

- 一般也无法解决组合型歧义问题

  例：做完作业才能看电视

  W1= 做/完/作业/才能/看/电视/

  W2= 做/完/作业/才/能/看/电视/

  P(W1) > P(W2)

分词方法

1. 基于词典的分词方法

a. 最大匹配法

b. 最优路径法
   - 1）词数最少的路径最优
   - 2）半词法
   - 3）最大概率法分词

2. 基于字序列标注的分词方法

4.3 基于字序列标注的分词方法

字位标注法

- 分词可以看做是对字加“词位标记”的过程

字位标注法

- 分词可以看做是对字加“词位标记”的过程
- “人”的词位分类示例：

B	E	M	S
词首	词尾	词中	独立词
人们	古人	小人国	听 人 说

基于字序列标注的方法

- 字位标注的原理：根据字本身及其上下文的特征，来决定当前字的词位标注

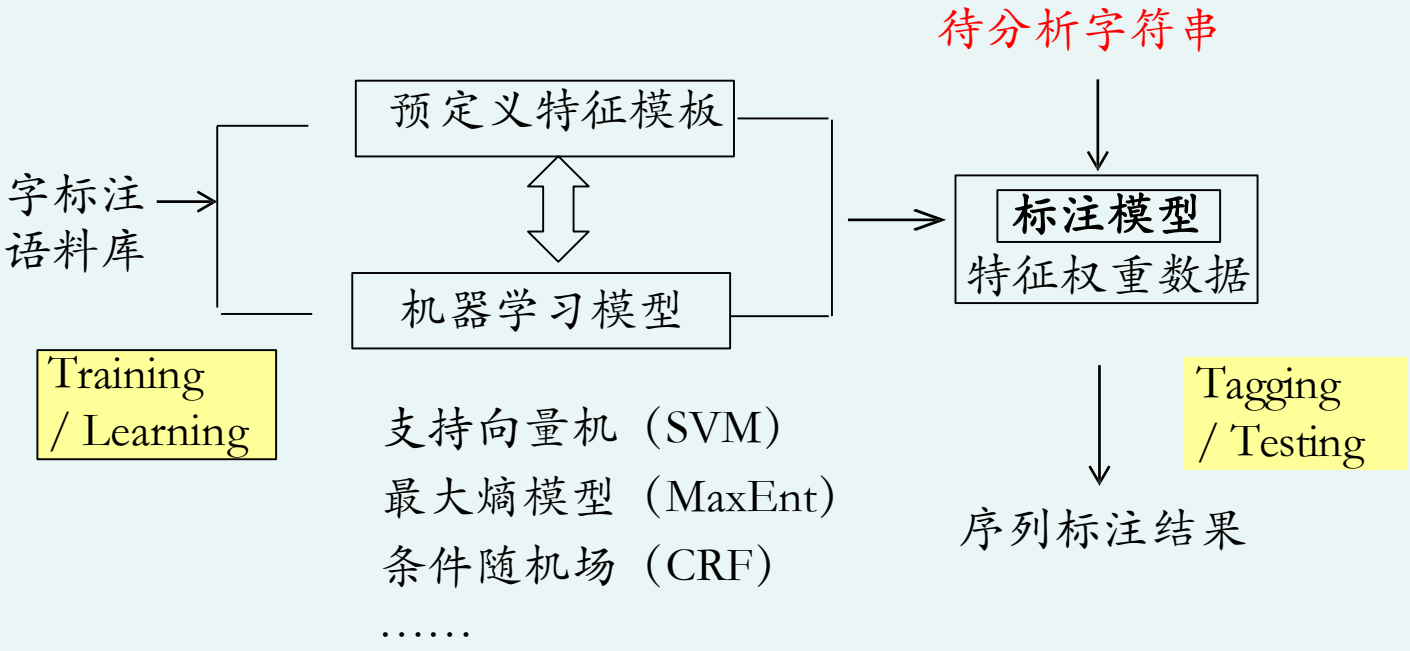
特征模板示例	含义
C_0	当前字
C_{-2}, C_{-1}, C_1, C_2	当前字的左边第二字，第一字，右边第一字，第二字
$C_{-1}C_0, C_0C_1$ $C_{-2}C_{-1}, C_1C_2$	当前字跟其左边一个字，当前字跟其右一个字 当前字的左边两个字，当前字的右边两个字
$C_{-1}C_1$	当前字的左边一个字加右边一个字
T_{-1} T_{-2}	左边第一个字的字位标注 左边第二个字的字位标注
Default feature	缺省特征（当上述特征都不适用时）

基于字序列标注的方法

自然句形式	已结婚的和尚未结婚的都应该到计生办登记
词切分结果	已/ 结婚/ 的/ 和/ 尚未/ 结婚/ 的/ 都/ 应该/ 到/ 计生办/ 登记/
字标注结果	已 结 婚 的 和 尚 未 结 婚 的 都 应 该 到 计 生 办 登 记 S B E S S B E B E S S B E S B M E B E

C_0 生成的特征	$C_{-1}C_0$生成的特征	C_0C_1生成的特征
和	的和	和尚
尚	和尚	尚未
未	尚未	未结
结	未结	结婚
婚	结婚	婚的
的	婚的	的都

基于字序列标注的方法

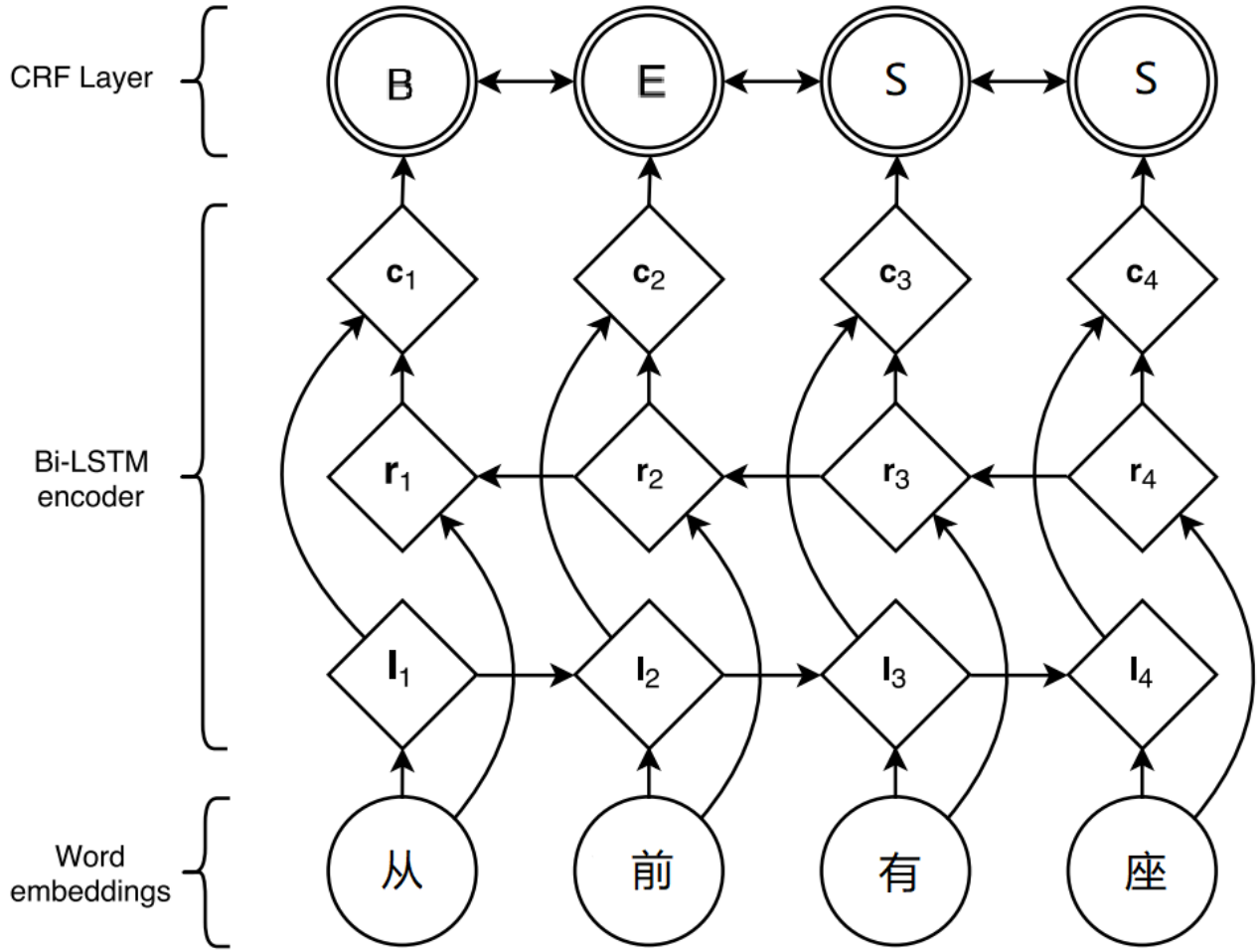


CRF 工具包： http://crfpp.googlecode.com/svn/trunk/doc/index.html

Maximum Entropy工具包： https://github.com/lzhang10/maxent

SVM 工具包： http://www.svms.org/software.html

基于字序列标注的方法



基于LSTM+CRF的序列标注方法

基于字序列标注的方法的优点

- 能够平衡地看待词表词和未登录词的识别问题。文本中的词表词和未登录词都是用统一的字标注来实现的
- 在学习架构上，既可以不必专门强调词表词信息，也不用专门设计特定的未登录词(如人名、地名、机构名)识别模块，这使得分词系统的设计大大简化
- 在字标注过程中，所有的字根据预定义的特征进行词位特性的学习，获得一个概率模型。然后，在待分字串上，根据字与字之间的结合紧密程度，得到一个词位的标注结果
- 在这样一个分词过程中，分词成为字重组的简单过程，结果令人满意的

基于字序列标注的方法的优点

- 能够平衡地看待词表词和未登录词的识别问题。文本中的词表词和未登录词都是用统一的字标注来实现的
- 在学习架构上，既可以不必专门强调词表词信息，也不用专门设计特定的未登录词(如人名、地名、机构名)识别模块，使得分词系统的设计大大简化
- 在字标注过程中，通过为字提供词语内部构成模式的特征进行词位特征的学习，获得一个统计模型。然后，在待分字串上，根据字与字之间的结合紧密程度，得到一个词位的标注结果
- 在这样一个分词过程中，分词成为字重组的简单过程，结果令人满意的

简单、鲁棒性强、效果好！！

4.4 中文分词技术的评测

中文分词技术的评测

- 计算分词正确率的不同标准
  - 1) 以词数算
  - 2) 以句数算

中文分词技术的评测

- 计算分词正确率的不同标准
  1) 以词数算
  2) 以句数算

- 分词质量对NLP应用系统的影响
  1) 分词质量对MT的影响
  2) 分词质量对IR的影响
  ……

准确率、召回率、F-Score

- 准确率(precision)

$$准确率（P）= \frac{切分结果中正确分词数}{切分结果中所有分词数} * 100\%$$

准确率、召回率、F-Score

- 准确率(precision)

$$准确率（P）= \frac{切分结果中正确分词数}{切分结果中所有分词数} * 100\%$$

- 召回率(recall)

$$召回率（R）= \frac{切分结果中正确分词数}{标准答案中所有分词数} * 100\%$$

准确率、召回率、F-Score

- 准确率(precision)

$$准确率（P）= \frac{切分结果中正确分词数}{切分结果中所有分词数} * 100\%$$

- 召回率(recall)

$$召回率（R）= \frac{切分结果中正确分词数}{标准答案中所有分词数} * 100\%$$

准确率、召回率、F-Score

- F-评价(F-measure 综合准确率和召回率的评价指标)

$$F1 = \frac{2*P*R}{P + R}$$

Thank you!

权小军 中山大学数据科学与计算机学院