信息检索导论--
逆向最大匹配算法实现

授课人：李永可
邮箱：lyk@xjau.edu.cn

正向最大匹配算法

最大匹配是指以词典为依据，取词典中最长单词长度Maxlen为第一个次取字数量的扫描串，在词典中进行扫描，如果在词典中发现匹配字符串，则匹配成功，当前字符串即为分词结果，若没有匹配字符串，则将当前处理字符串最右侧字去掉，然后重新开始一轮匹配，直到找到匹配字符串为至，若没有一个字符串能够匹配，则截取待匹配字符串第一个字符为匹配结果。

正向最大匹配算法

- 以“我们在野生动物园玩”详细说明一下正向最大匹配方法：

正向即从前往后取词，从7->1，每次减一个字，直到词典命中或剩下1个单字。

第1次：“我们在野生动物”，扫描词典，无

第2次：“我们在野生动”，扫描词典，无

。。。。

第6次：“我们”，扫描词典，有

扫描中止，输出第1个词为“我们”，去除第1个词后开始第2轮扫描

正向最大匹配算法

- 第2轮扫描：
- 第1次：“在野生动物园玩”，扫描词典，无
- 第2次：“在野生动物园”，扫描词典，无
- 。 。 。 。
- 第6次：“在野”，扫描词典，有
- 扫描中止，输出第2个词为“在野”，去除第2个词后开始第3轮扫描

正向最大匹配算法

- 第3轮扫描：
- 第1次：“生动物园玩”，扫描词典，无
- 第2次：“生动物园”，扫描词典，无
- 第3次：“生动物”，扫描词典，无
- 第4次：“生动”，扫描词典，有
- 扫描中止，输出第3个词为“生动”，第4轮扫描

正向最大匹配算法

- 第4轮扫描：
- 第1次：“物园玩”，扫描词典，无
- 第2次：“物园”，扫描词典，无
- 第3次：“物”，扫描词典，无
- 扫描中止，输出第4个词为“物”，非字典词数加1，开始第5轮扫描

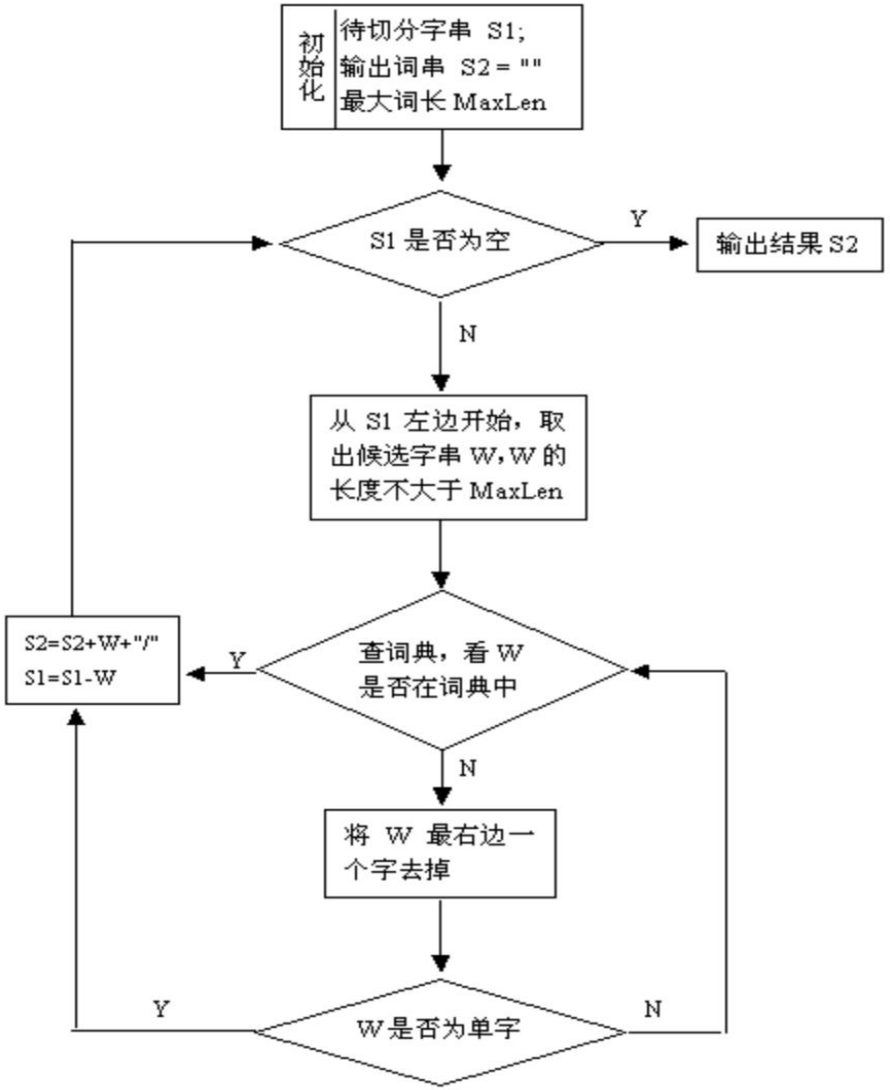
正向最大匹配算法

- 第5轮扫描：
- 第1次：“园玩”，扫描词典，无
- 第2次：“园”，扫描词典，有
- 扫描中止，输出第5个词为“园”，单字字典词数加1，开始第6轮扫描

正向最大匹配算法

- 第6轮扫描：
- 第1次：“玩”，扫描字典，有
- 扫描中止，输出第6个词为“玩”，单字字典词数加1，整体扫描结束。
- 正向最大匹配法，最终切分结果为：“我们/在野/生动/物/园/玩”，其中，单字字典词为2，非词典词为1。

正向最大匹配算法


初始化
待切分字串 S1;
输出词串 S2 = ""
最大词长 MaxLen
S1 是否为空
Y
输出结果 S2
N
从 S1 左边开始，取出候选字串 W，W 的长度不大于 MaxLen
查词典，看 W 是否在词典中
Y
S2=S2+W+"/"
S1=S1-W
N
将 W 最右边一个字去掉
W 是否为单字
Y
N

正向最大匹配算法

- 参考网页：
- http://yangshangchuan.iteye.com/blog/2031813

正向最大匹配算法实现步骤

- 1.创建java工程文件Mm
- 2.加载字典文件。编写加载字典文件方法loaddic（String path）
- 3.利用正向最大匹配算法编写分词函数fenci(String words)

逆向最大匹配算法

- 逆向即从后往前取词，其他逻辑和正向相同。即：
- 第1轮扫描：“在野生动物园玩”
- 第1次：“在野生动物园玩”，扫描词典，无
- 第2次：“野生动物园玩”，扫描词典，无
- 。 。 。 。
- 第7次：“玩”，扫描1字词典，有
- 扫描中止，输出“玩”，单字字典词加1，开始第2轮扫描

逆向最大匹配算法

- 第2轮扫描：“们在野生动物园”
- 第1次：“们在野生动物园”，扫描词典，无
- 第2次：“在野生动物园”，扫描词典，无
- 第3次：“野生动物园”，扫描词典，有
- 扫描中止，输出“野生动物园”，开始第3轮扫描

逆向最大匹配算法

- 第3轮扫描：“我们在”
- 第1次：“我们在”，扫描3字词典，无
- 第2次：“们在”，扫描2字词典，无
- 第3次：“在”，扫描1字词典，有
- 扫描中止，输出“在”，单字字典词加1，开始第4轮扫描

逆向最大匹配算法

- 第4轮扫描：“我们”
- 第1次：“我们”，扫描2字词典，有
- 扫描中止，输出“我们”，整体扫描结束。
- 逆向最大匹配法，最终切分结果为：“我们/在/野生动物园/玩”，其中，单字字典词为2，非词典词为0。