中文分词

# 基本算法

## 基于词典的分词

最大正向匹配法(MM, MaximumMatching Method,MM)

（１）求出用于分词的词典里的最长的词的词长，长度标为ｎ。

（２）从一个句子的左边开始截取一个长度为ｎ的字符串，用来和词典中的词进行依次比较。

（３）如果找到这样的词，就匹配成功，将该词从句子中切分出来。

（４）如果没找到这样的词，则把该字符串的最后边的字截去，用剩下的ｎ－１字长的字符串继续从词典中进行查找，查找成功则从句子中切分出来，不成功就继续截去最后边的字用剩下的ｎ－２字长的字符串进行继续查找匹配。

（５）当ｎ＝１时说明词典中没有这样的词，舍去，从下一个字重新开始正向最大匹配查找，直到整个句子切分完成。从上述的方法发现正向最大匹配方法原理很简单，也便于实现。但是在判别歧义上往往导致切分的错误。据资料统计，正向最大匹配方法的错误切分率是１／１６９。

逆向最大匹配法(ReverseMaximum Matching Method,RMM)

反向最大匹配方法也是很常用的分词方法，它和正向最大

匹配方法的思想很类似，只不过正向最大匹配方法是从句子的

左边开始匹配，不成功时首先截去字符串的最右边的字，而反向

最大匹配是从句子的右边开始匹配，不成功时首先截取字符串

的最左边的字

下图为最大正向匹配法算法流程图：

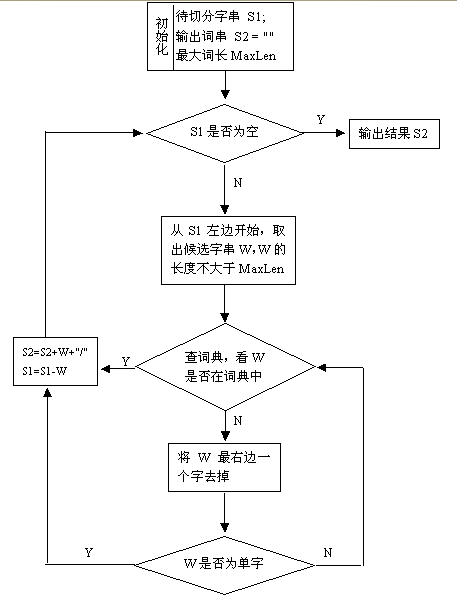


图1最大正向匹配法算法流程图

**N-gram模型**

该模型基于这样一种假设，第n个词的出现只与前面N-1个词相关，而与其它任何词都不相关，整句的概率就是各个词出现概率的乘积。我们给定一个词，然后猜测下一个词是什么。N-gram模型的主要思想就是这样的。



图 2 N-gram公式

前边已经利用前向最大匹配和后向最大匹配对句子进行了中文分词。为了提高分词的准确度和歧义处理能力，可以利用N-gram比较前向、后向哪个分词的得到的概率结果更大，就取相应的分词结果,这样就能出来句子歧义。

**隐马尔科夫模型(Hidden Markov Model, HMM)**

由于基于频度和词典的分词会倾向于把不能识别的词组一个字一个字地切割开，所以对这些字的合并就是识别OOV的一个方向，定义一个buf 变量收集了这些连续的单个字，最后把它们组合成字符串再交由 HMM来进行下一步分词。HMM可以处理OOV及歧义词

**HMM** **模型介绍**

StatusSet: 状态值集合S={B,M,E,S} ObservedSet: 观察值集合

TransProbMatrix: 转移概率矩阵 EmitProbMatrix: 发射概率矩阵

InitStatus: 初始状态分布

**HMM解码问题(预测问题)**   
 即给定观测序列 O=O1,O2,O3…Ot和模型参数λ=(A,B,π)，怎样寻找满足这种观察序列意义上最优的隐含状态序列S。(viterbi算法，S={B,M,E,S})

**Viterbi的算法步骤如下：**

步骤１ 计算第一个字的概率：状态的初始概率乘以隐藏状态到观察状态的条件概率。

步骤２ 计算第一个字以后的概率，首先从前一个字的每个状态转移到当前状态的概率的最大值，然后乘以隐藏状态到观察状态的条件概率。

步骤３ 找到最后一个字呈现哪种状态的概率最大。

步骤４ 用动态规划的方法逆推回去各个字出现什么状态概率最大。

# 处理歧义及OOV能力

1. 利用前向最大匹配和后向最大匹配对句子进行了中文分词。当两者分词结果不一致时即出现了歧义，为了提高分词的准确度和歧义处理能力，可以利用N-gram比较前向、后向哪个分词的得到的概率结果更大，就取相应的分词结果,这样就能出来句子歧义。
2. 由于基于频度和词典的分词会倾向于把不能识别的词组一个字一个字地切割开，所以对这些字的合并就是识别OOV的一个方向，定义一个buf变量收集了这些连续的单个字，最后把它们组合成字符串再交由 HMM来进行下一步分词。HMM可以处理OOV及歧义词

# 评估方法

常用的评测指标包括准确率P(Precision)、召回率R(Recall)和F值(F Score)。

P=分词结果中切分正确的总词数/分词结果中的总词数

R=分词结果中切分正确的总词数/标准结果中的总词数

F=2PR/P+R

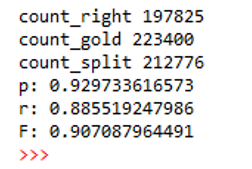


图3 评估结果

图2为通过N-gram分词后，不在词典中的未处理的单个连续字组成的字符串进行分词，提供OOV及处理歧义能力。

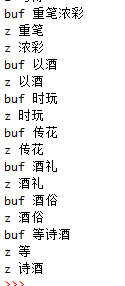


图4HMM分词