中文分词-01

OutLine

- 中文分词基础
- Jieba分词
- 【基础实践】分词实践

背景

- 一段文字不仅仅在于字面上是什么，还在于怎么切分和理解。
- 例如：
  - 阿三炒饭店：
  - 阿三 / 炒饭 / 店　　阿三 / 炒 / 饭店
- 和英文不同，中文词之间没有空格，所以实现中文搜索引擎，比英文多了一项分词的任务。
- 如果没有中文分词会出现：
  - 搜索“达内”，会出现“齐达内”相关的信息

背景

- 要解决中文分词准确度的问题，是否可以提供一个免费版本的通用分词程序?
  - 像分词这种自然语言处理领域的问题，很难彻底完全解决
  - 每个行业或业务侧重不同，分词工具设计策略也是不一样的

切分方案

- 切开的开始位置对应位是1，否则对应位是0，来表示“有/意见/分歧”的bit内容是：11010
- 还可以用一个分词节点序列来表示切分方案，例如“有/意见/分歧”的分词节点序列是{0,1,3,5}

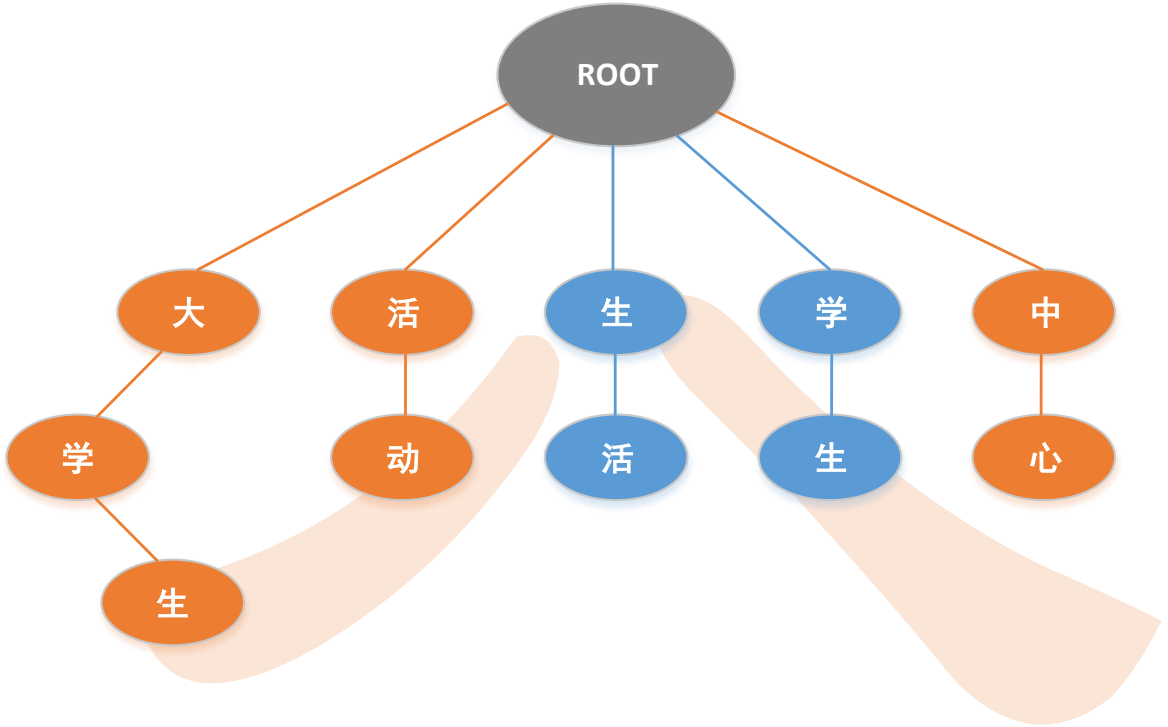
最常见方法

- 最常见的分词方法是基于词典匹配
  - 最大长度查找（前向查找，后向查找）

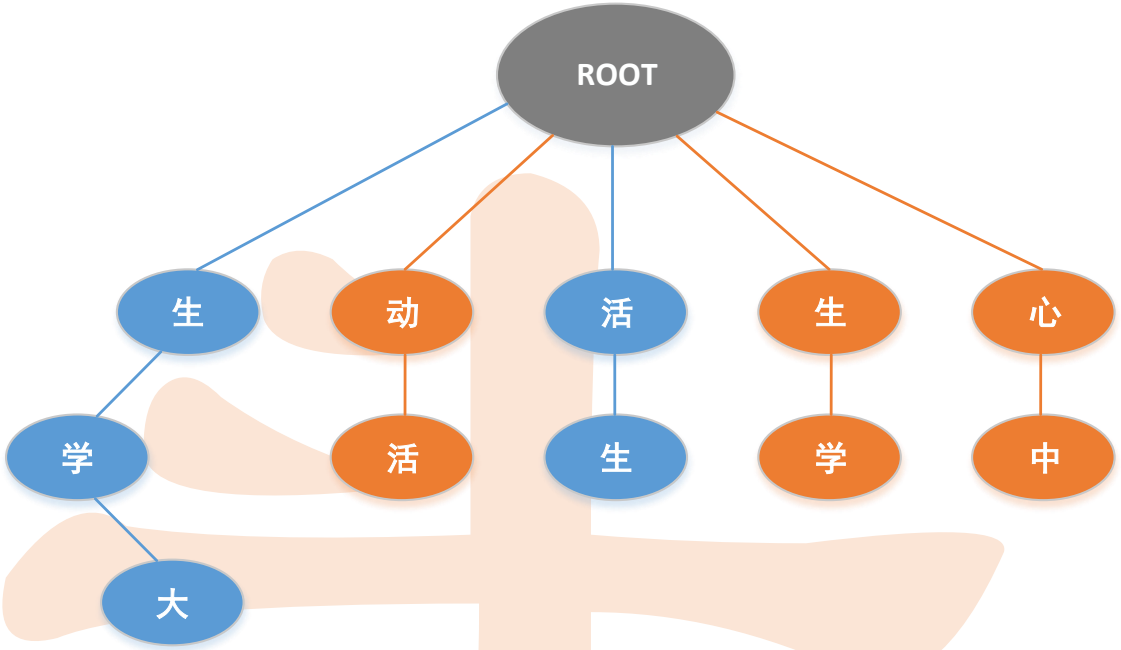
- 数据结构
  - 为了提高查找效率，不要逐个匹配词典中的词
  - 查找词典所占的时间可能占总的分词时间的1/3左右，为了保证切分速度，需要选择一个好的查找词典方法
  - Trie树常用于加速分词查找词典问题

Trie树

- 例如：大学生活动中心



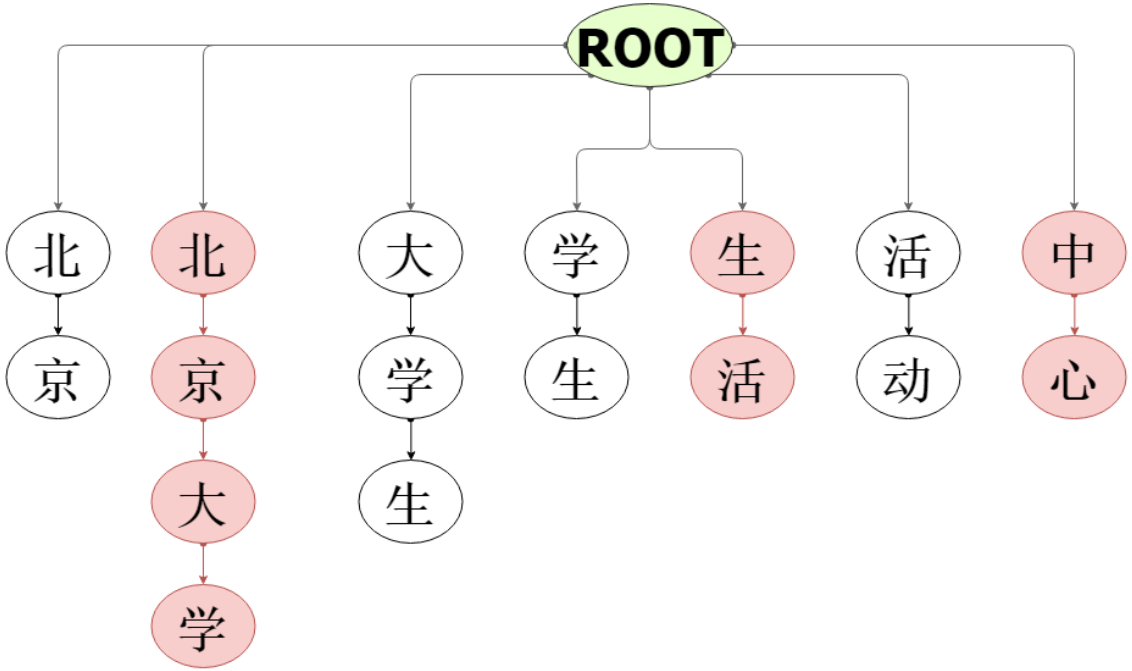
正向：大学生 / 活动 / 中心

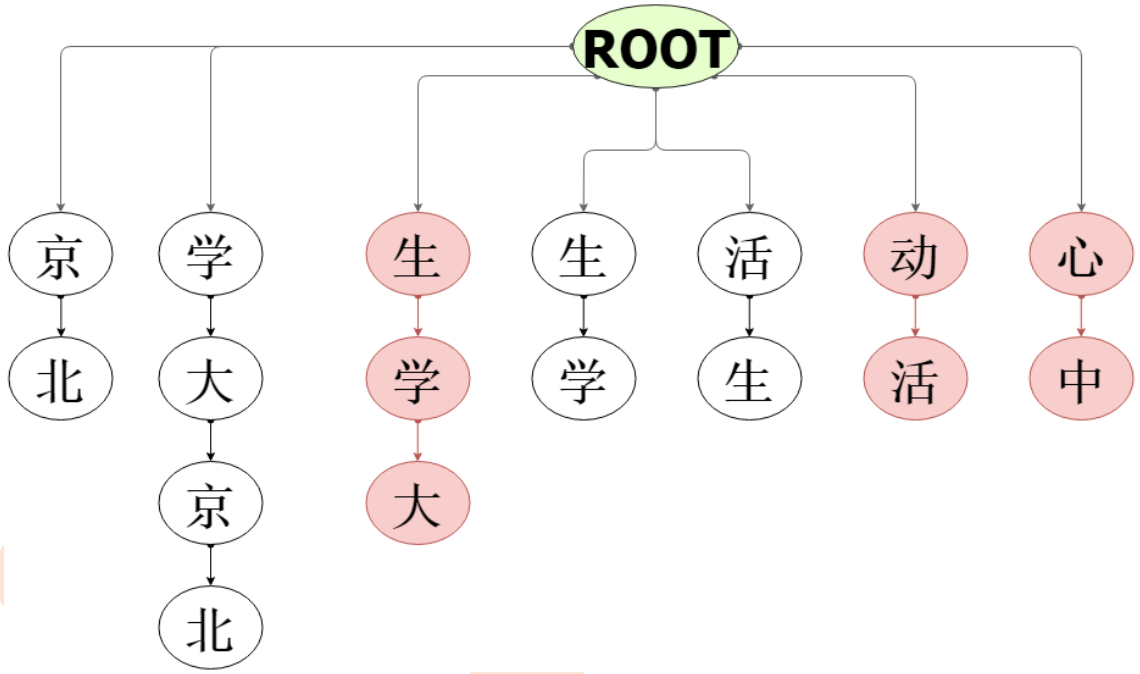


反向：大学生 / 活动 / 中心

Trie树

- 例如：北京大学生活动中心



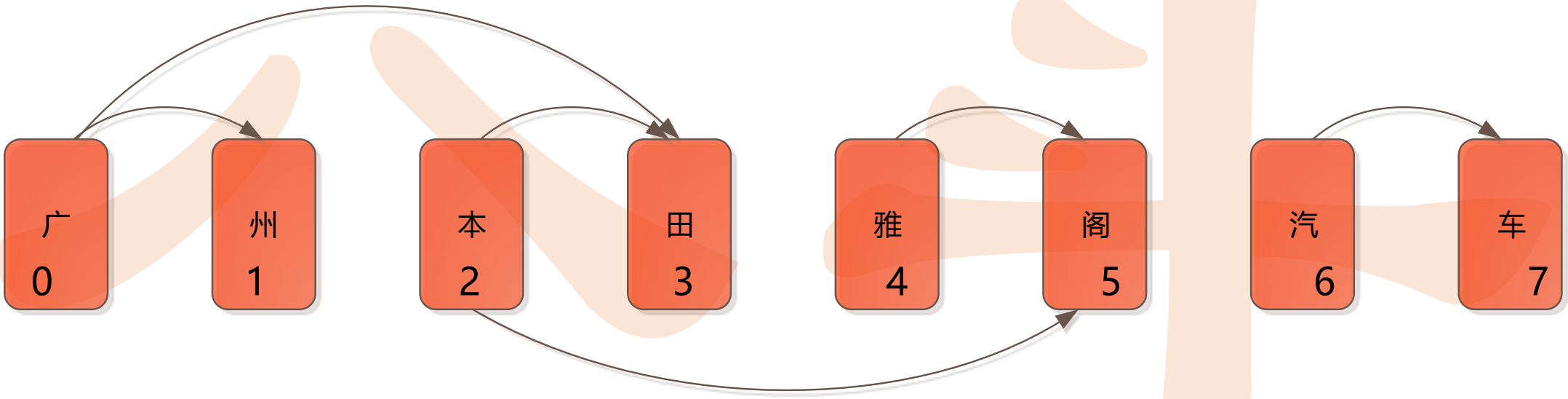


正向：北京大学 / 生活 / 动 / 中心

反向：北京 / 大学生 / 活动 / 中心

切 分 词 图

广州本田雅阁汽车 → DAG: {0: [0, 1, 3], 1: [1], 2: [2, 3, 5], 3: [3], 4: [4, 5], 5: [5], 6: [6, 7], 7: [7]}



概率语言模型

- 假设需要分出来的词在语料库和词表中都存在，最简单的方法是按词计算概率，而不是按字算概率。
- 从统计思想的角度来看，分词问题的输入是一个字串C=c1,c2……cn，输出是一个词串S=w1,w2……wm，其中m<=n。对于一个特定的字符串C，会有多个切分方案S对应，分词的任务就是在这些S中找出一个切分方案S，使得P(S|C)的值最大。
- P(S|C)就是由字符串C产生切分S的概率，也就是对输入字符串切分出最有可能的词序列。

$$Seg(C) = \arg\max_{S \in G} P(S \mid C) = \arg\max_{S \in G} \frac{P(C \mid S)P(S)}{P(C)}$$

例 子

- 例如：对于输入字符串C“南京市长江大桥”，有下面两种切分可能：
  - S1：*南京市 / 长江 / 大桥*
  - S2：南京 / 市长 / 江大桥
- 这两种切分方法分别叫做S1和S2。计算条件概率P(S1|C)和P(S2|C)，然后根据P(S1|C)和P(S2|C)的值来决定选择S1还是S2。
- P(C)是字串在语料库中出现的概率。比如说语料库中有1万个句子，其中有一句是 “南京市长江大桥”那么P(C)=P(“南京市长江大桥”)=万分之一。
- 因为P(C∩S) = P(S|C)*P(C) = P(C|S)*P(S)，所以 $$P(S|C)=\frac{P(C|S)\times P(S)}{P(C)}$$

例 子

- 贝叶斯公式：$P(S \mid C) = \frac{P(C \mid S) \times P(S)}{P(C)}$
- P(C)只是一个用来归一化的固定值
- 另外：从词串恢复到汉字串的概率只有唯一的一种方式，所以P(C|S)=1。
- 所以：比较P(S1|C)和P(S2|C)的大小变成比较P(S1)和P(S2) 的大小

$$\frac{P(S_1 \mid C)}{P(S_2 \mid C)} = \frac{P(S_1)}{P(S_2)}$$

- 因为P(S1)=P(南京市,长江,大桥)=P(南京市)*P(长江)*P(大桥) > P(S2)=P(南京,市长,江大桥)，所以选择切分方案S1。

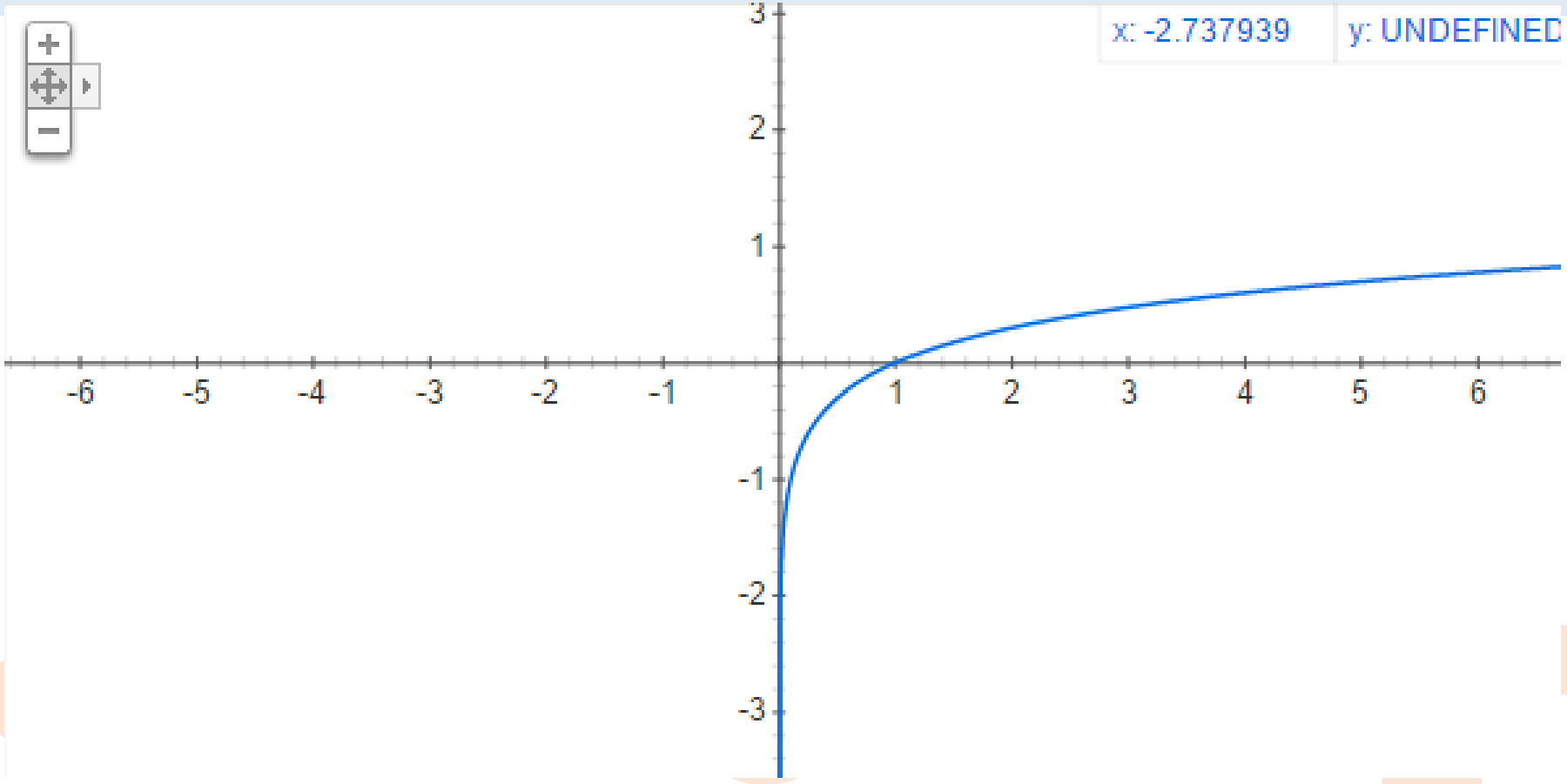
例 子

- 为了容易实现，假设每个词之间的概率是上下文无关的，则：

$$P(S)=P(w_1,w_2,...,w_m)\approx P(w_1)\times P(w_2)\times...\times P(w_m)\propto \log P(w_1)+\log P(w_2)+...+\log P(w_m)$$

- 其中，P(w) 就是这个词出现在语料库中的概率。因为函数y=log(x)，当x增大，y也会增大，所以是单调递增函数。∝是正比符号。因为词的概率小于1，所以取log后是负数。
- 最后算 logP(w)。取log是为了防止向下溢出，如果一个数太小，例如0.00000000000000000000000000001 可能会向下溢出。
- 如果这些对数值事前已经算出来了，则结果直接用加法就可以得到，而加法比乘法速度更快。

Log图形化表示



log(0.00000000000000000000000000001) = -30

一元模型

- 对于不同的S，m的值是不一样的，一般来说m越大，P(S)会越小。也就是说，分出的词越多，概率越小。

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

- 因此：$\log P(w_i) = \log(Freq_w) - \log N$
- 这个P(S)的计算公式也叫做基于一元模型的计算公式，它综合考虑了切分出的词数和词频。

N 元 模 型

- 假设在日本，[和服]也是一个常见的词。按照一元概率分词，可能会把“产品和服务”分成[产品][和服][务]。为了切分更准确，要考虑词所处的上下文。
- 给定一个词，然后猜测下一个词是什么。当我说“NBA”这个词时，你想到下一个词是什么呢？我想大家有可能会想到“篮球”，基本上不会有人会想到“足球”吧。
- 之前为了简便，所以做了“前后两词出现概率是相互独立的”的假设在实际中是不成立的

N 元 模 型

- N元模型使用n个单词组成的序列来衡量切分方案的合理性：
- 估计单词w1后出现w2的概率。根据条件概率的定义： $P(w_2 \mid w_1) = \frac{P(w_1, w_2)}{P(w_1)}$
- 可以得到：P(w1,w2)= P(w1)P(w2|w1)
- 同理：P(w1,w2,w3)= P(w1,w2)P(w3|w1,w2)
- 所以有：P(w1,w2,w3)= P(w1)P(w2|w1)P(w3|w1,w2)
- 更加一般的形式：
- P(S)=P(w1,w2,...,wn)= P(w1)P(w2|w1)P(w3|w1,w2)...P(wn|w1w2...wn-1)
- 这叫做概率的链规则。

N 元 模 型

- 如果简化成一个词的出现仅依赖于它前面出现的一个词，那么就称为二元模型(Bigram)。
- $P(S) = P(w1,w2,...,wn) = P(w1)\ P(w2|w1)\ P(w3|w1,w2)...P(wn|w1w2...wn-1)$

  $\approx P(w1)\ P(w2|w1)P(w3|w2)...P(wn|wn-1)$
- 如果简化成一个词的出现仅依赖于它前面出现的两个词，就称之为三元模型(Trigram)。
- 如果一个词的出现不依赖于它前面出现的词，叫做一元模型(Unigram)

OutLine

- 中文分词基础
- Jieba分词
- 【基础实践】分词实践

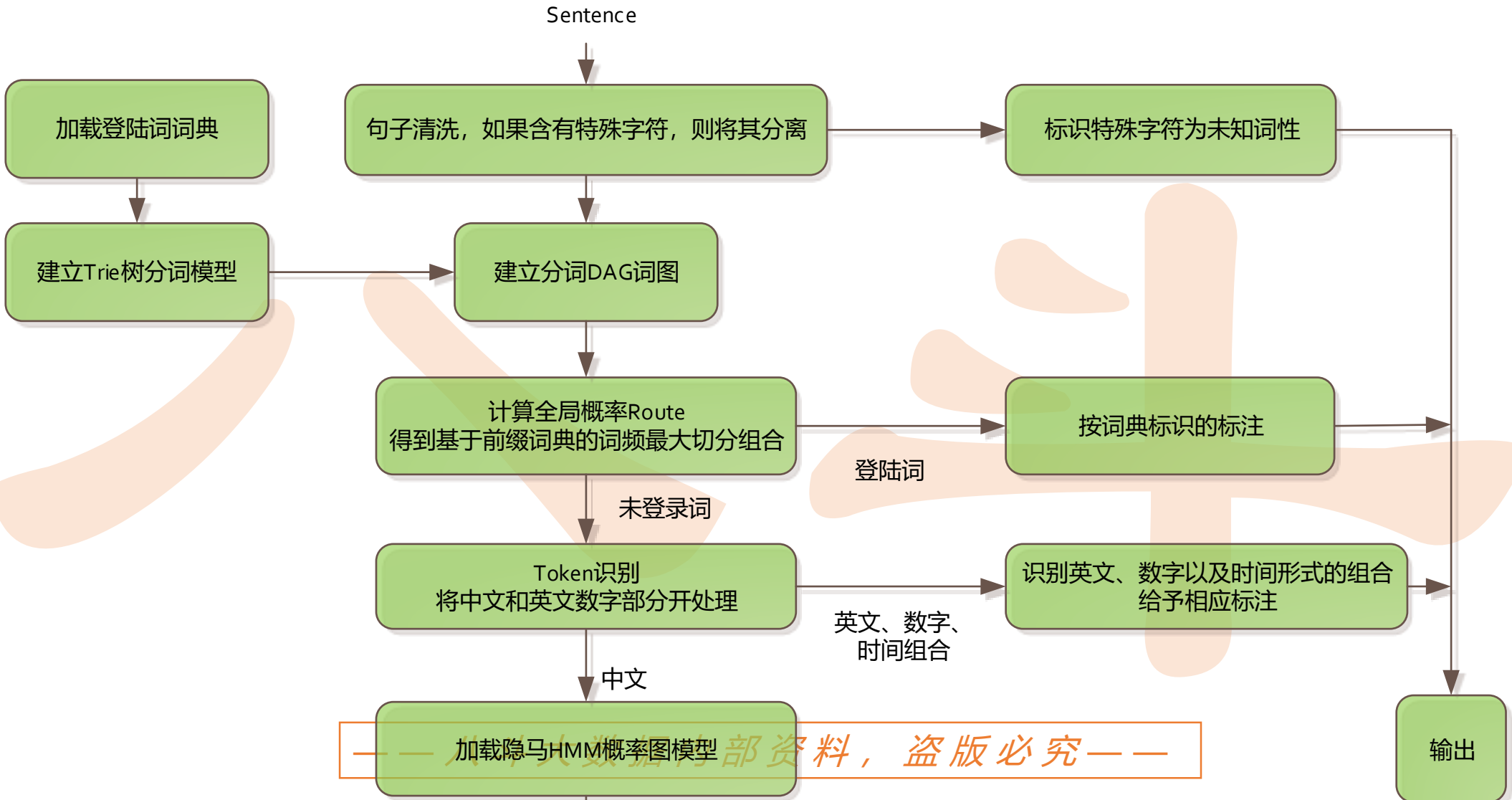
Jieba分词简介

- 源码下载的地址：https://github.com/fxsjy/jieba
- 支持三种分词模式
  - 精确模式：将句子最精确的分开，适合文本分析
  - 全模式：句子中所有可以成词的词语都扫描出来，速度快，不能解决歧义
  - 搜索引擎模式：在精确模式基础上，对长词再次切分，提高召回
- 支持繁体分词
- 支持自定义字典

Jieba分词简介

- 基于Trie树结构实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图（DAG)
- 采用了动态规划查找最大概率路径, 找出基于词频的最大切分组合
- 对于未登录词，采用了基于汉字成词能力的HMM模型，使用了Viterbi算法

Jieba分词细节

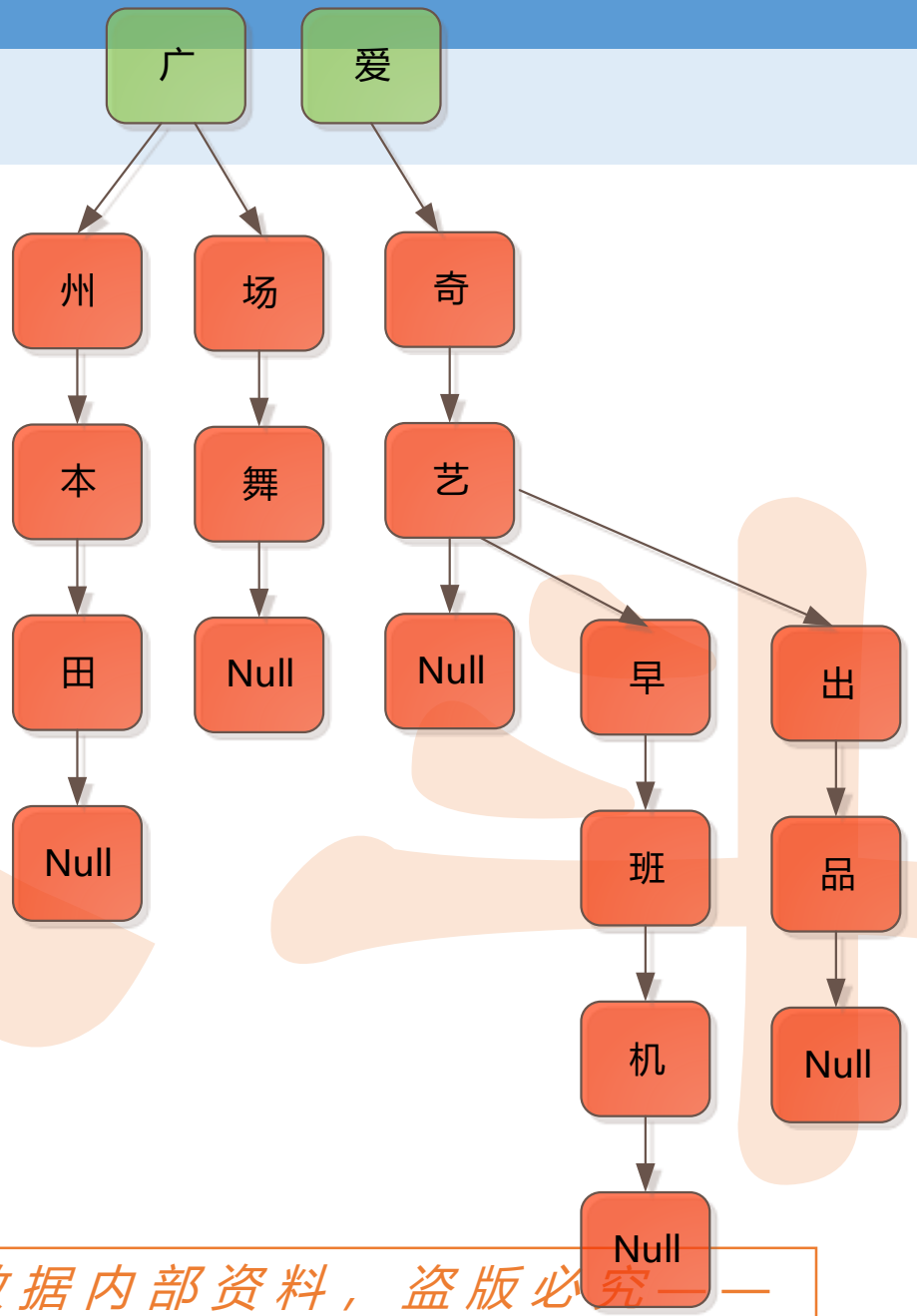


Jieba登录词词库加载

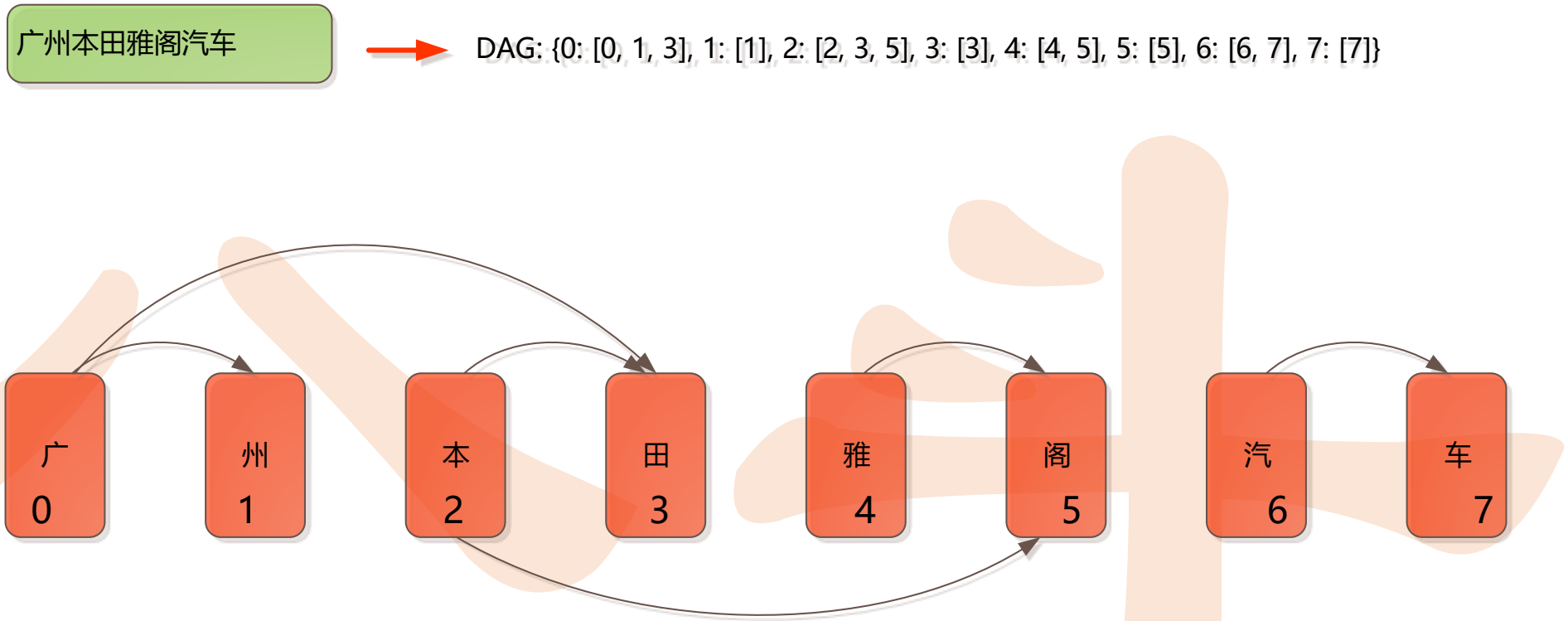
词表

```
广州本田 2 n
广场舞 2290 n
爱奇艺 455 n
爱奇艺早班机 2400 n
爱奇艺出品 270 n
```

生成trie树



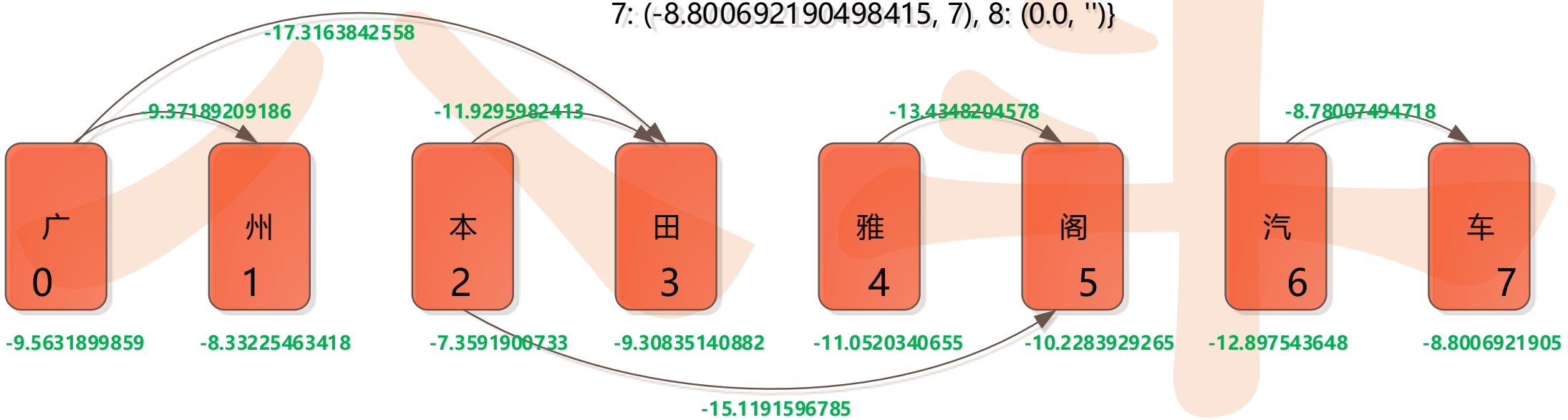
Jieba的DAG词图



Jieba的Route概率-获得词频最大切分

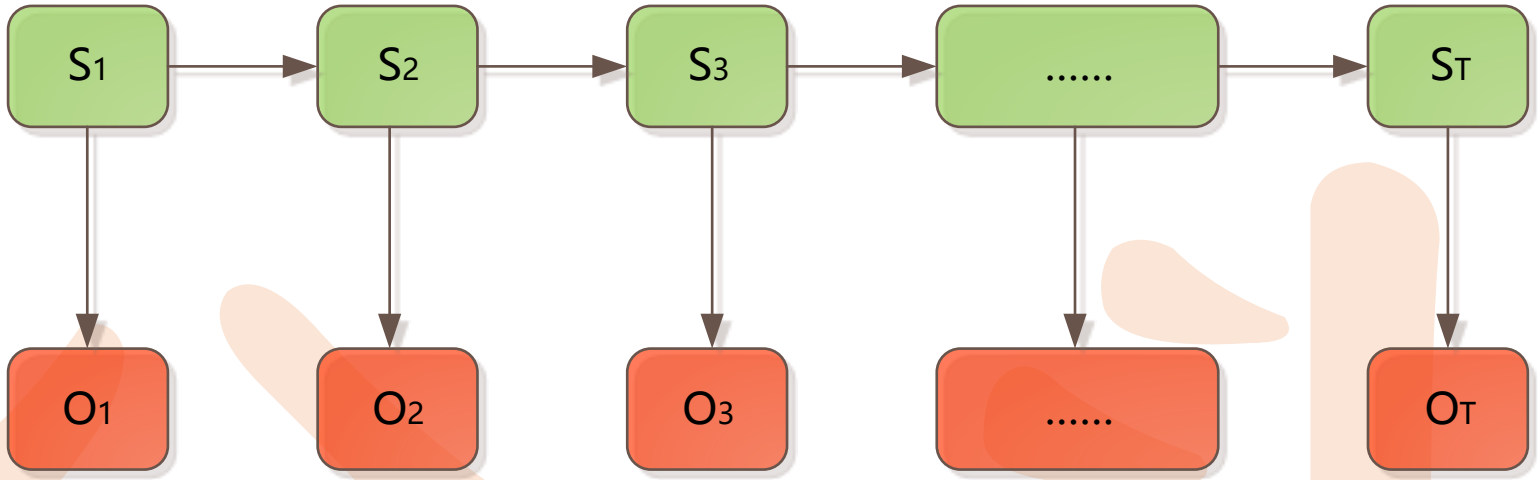
广州本田雅阁汽车

route: {0: (-33.271126717488308, 1),
1: (-32.231489259807965, 1),
2: (-23.899234625632083, 5),
3: (-31.523246813843 94, 3),
4: (-22.214895405024865, 5),
5: (-19.00846787368323, 5),
6: (-8.7800749471799175, 7),
7: (-8.800692190498415, 7), 8: (0.0, '')}



隐马尔可夫模型

- 观察和隐藏序列共同构成隐马尔可夫模型（HMM）



- O($o_1 o_2 \ldots o_T$)：观测序列，o_t只依赖于s_t
- S($s_1 s_2 \ldots s_T$)：状态序列（隐藏序列），S是Markov序列，假设1阶Markov序列，则s_{t+1}只依赖于s_t

Q & A

@八斗学院