

0序列标注

序列标注简单的来说就是给定一个序列，对序列中的每一个元素做一个标记，或者说给每一个元素打一个标签，这是一个比较宽泛的概念。`中文命名实体识别、中文分词和词性标注`等这些基本的NLP任务都属于序列标注的范畴。

1命名实体识别

今年海钓比赛在厦门市与金门之间的海域举行。

标注结果：

今(O)年(O)海(O)钓(O)比(O)赛(O)在(O)厦(B-LOC)门(I-LOC)市(E-LOC)与(O)金(B-LOC)门(E-LOC)之(O)间(O)的(O)海(O)域(O)举(O)行(O)。(O)

上述采用的是BIO的标注规范，此外还有BIOE和BIOES两种规范

例子：

他	和	爸爸	去	电影	院	看	哈利	波特
B-Per	O	B-Per	O	B-Loc	I-Loc	O	B-Per	I-Per

他	和	爸	爸	去	电	影	院	看
B-Per	O	B-Per	E-Per	O	B-Loc	I-Loc	E-Loc	O
哈	利	波	特					
B-Per	I-Per	I-Per	E-Per					

他	和	爸	爸	去	电	影	院	看
S-Per	O	B-Per	I-Per	O	B-Org	I-Org	I-Org	O
哈	利	波	特					
B-Per	I-Per	I-Per	I-Per					

序列标注的训练集有哪些？

1、`自由时报`数据集

`人名、地名、组织名`三种实体类型。

2、`New York Times` 数据集

`NYT`数据集是通过将`freebase`中的关系与`纽约时报（NYT）`语料库对齐而成的。`纽约时报``New York Times`数据集包含150篇来自`纽约时报`的商业文章。抓取了从2009年11月到2010年1月`纽约时报`站上的所有文章。

3、`MSRA微软亚洲研究院`数据集

5万多条中`命名实体识别标注`数据（包括地点、机构、人物）

4、`CoNLL 2003`

这个数据集包括1393篇英语新闻文章和909篇德语新闻文章。英语语料库是免费的，德国语料库需要收钱(75美元)。英语语料实际上是`RCV1(Reuters Corpus, Volume 1, https://trec.nist.gov/data/reuters/reuters.html)`，路透社早些年公开的一些数据集。

In []:

目前有哪些工具？

1、HanLP

HanLP（汉语处理包）是款开源的使Java进开发的中自然语处理具，提供的功能包括中分词、词性标注、命名实体识别、依存句法分析等。

2、哈 LTP

LPT(Language Technology Platform)是哈尔滨工业学开发的中自然语处理具。代码开源，商业使付费。持中分词、词性标注、命名实体识别、依存句法分析、语标注（中）

3、清华 THU LAC

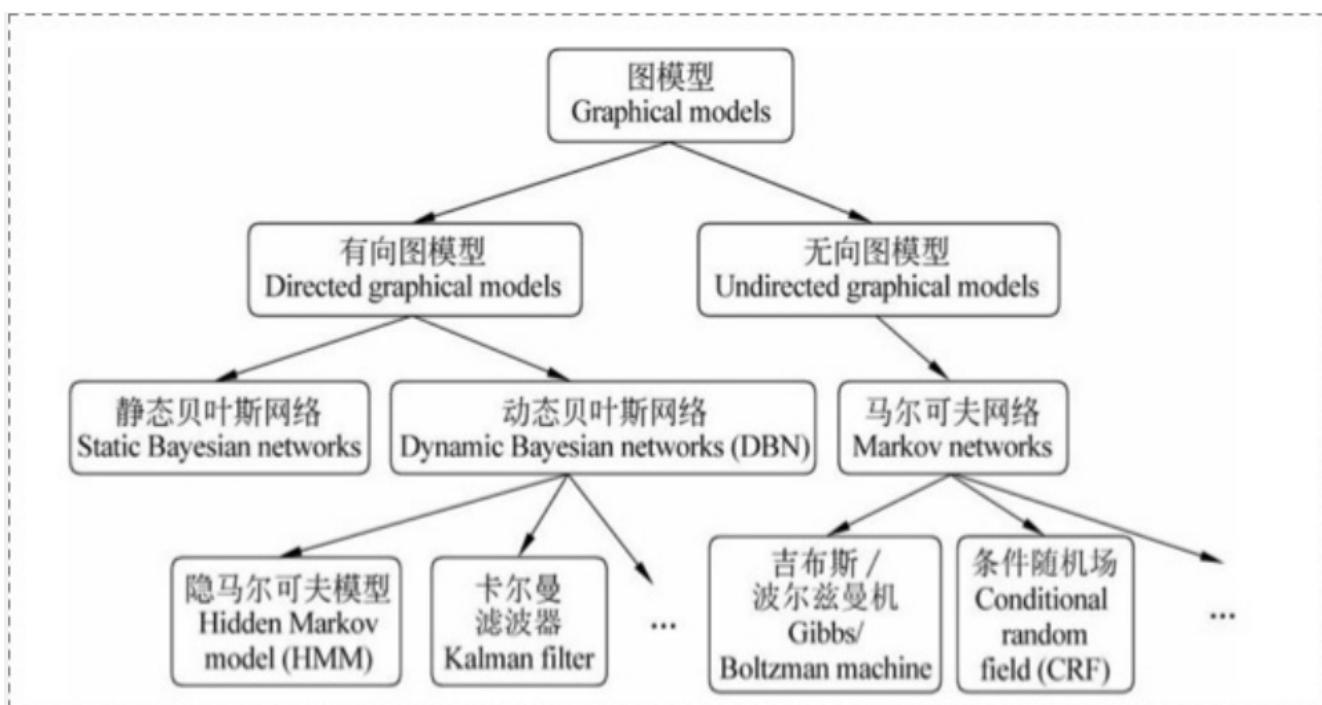
THULAC (THU Lexical Analyzer for Chinese) 是由清华学自然语处理与社会计算实验室研制推出的套中词法分析具包，具有中分词和词性标注功能。

4、斯坦福 Stanford CoreNLP

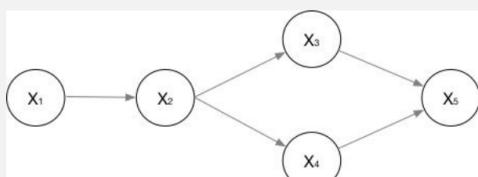
然语处理具包，能实现对然语文本的本分析，包括词形还原，词性标注、命名实体标注、共指消解、句法分析以及依存分析等功能

2 实现方法

HMM方法



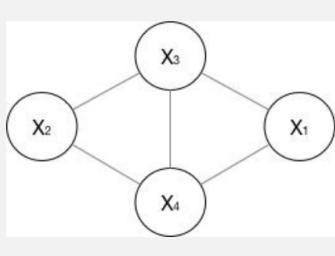
3.2、序列标注—有向图和无向图



$$\text{公式1} \quad P(x_1, \dots, x_n) = \prod_{i=0} P(x_i | \pi(x_i))$$

$$\text{公式2}$$

$$P(x_1, \dots, x_n) = P(x_1) \cdot P(x_2|x_1) \cdot P(x_3|x_2) \cdot P(x_4|x_2) \cdot P(x_5|x_3, x_4)$$



概率无向图模型，又称为马尔可夫随机场

它假设随机场中任意一个结点的赋值，仅仅和它的邻结点的取值有关，和不相邻的结点的取值无关。

无向图G中任何两个结点均有边连接的结点子集称为团。若C是无向图的一个团，且不能再加进任何一个G的结点使其成为更大的一个团，则此C为最大团。

$$\text{公式3} \quad P(Y) = \frac{1}{Z(x)} \prod_c \psi_c(Y_c)$$

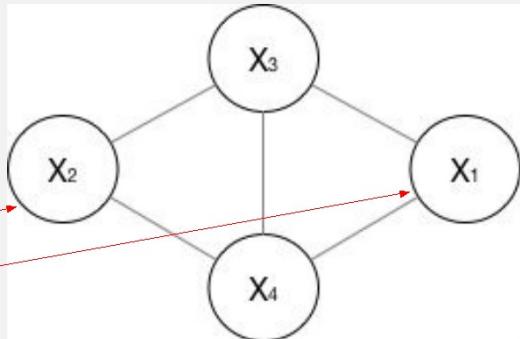
$$\text{公式4} \quad Z(x) = \sum_Y \prod_c \psi_c(Y_c)$$

$$\text{公式5} \quad \psi_c(Y_c) = e^{-E(Y_c)}$$

3.2、成对马尔可夫性

- 设 u 和 v 是无向图 G 中任意两个没有边连接的结点，结点 u 和 v 分别对应随机变量 Y_u 和 Y_v 。其他所有结点为 O （集合），对应的随机变量组是 Y_O 。成对马尔可夫性是指给定随机变量组 Y_O 的条件下随机变量 Y_u 和 Y_v 是条件独立的，其实意思就是说没有直连边的任意两个节点是独立的，即

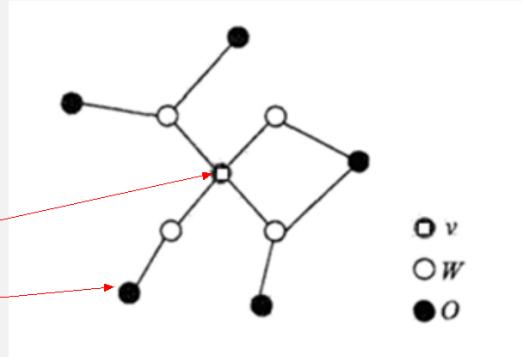
$$P(Y_u, Y_v | Y_O) = P(Y_u | Y_O)P(Y_v | Y_O)$$



3.2、局部马尔可夫性

- 设 $v \in V$ 是无向图 G 中任意一个结点， W 是与 v 有边连接的所有结点， O 是 v, W 以外的其他所有结点。 v 表示的随机变量是 Y_v , W 表示的随机变量组是 Y_W , O 表示的随机变量组是 Y_O 。局部马尔可夫性是指在给定随机变量组 Y_W 的条件下随机变量 v 与随机变量组 Y_O 是独立的，即

$$P(Y_v, Y_O | Y_W) = P(Y_v | Y_W)P(Y_O | Y_W)$$

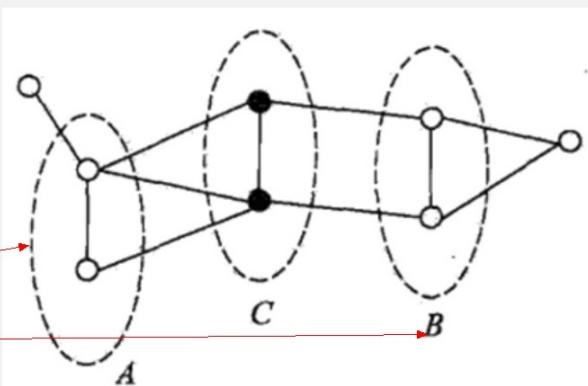


3.2、全局马尔可夫性

- 设结点集合 A, B 是在无向图 G 中被结点集合 C 分开的任意结点集合，如图所示。结点集合 A, B 和 C 所对应的随机变量组分别是 Y_A, Y_B 和 Y_C 。全局马尔可夫性是指给定随机变量组条件下随机变量组 Y_A 和 Y_B 是条件独立的，即

$$P(Y_A, Y_B | Y_C) = P(Y_A | Y_C)P(Y_B | Y_C)$$

A和B相互条件独立



3.2、序列标注

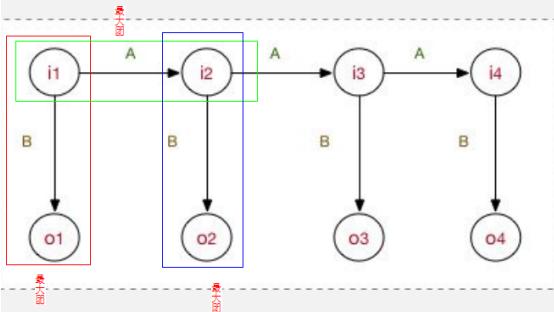
判别式 (discriminative) 模型 vs. 生成式(generative)模型

对于输入 x , 类别标签 y :

判别式模型常见的主要有：
Logistic Regression
SVM
CRF
Linear Regression

生成式模型常见的主要有：
Naive Bayes
HMMs
Markov Random Fields

3.2、HMM隐马尔可夫模型

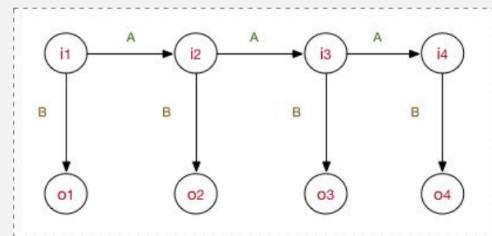


- 小明所在城市的天气有{晴天，阴天，雨天}三种情况，小明每天的活动有{宅，打球}两种选项。
- 作为小明的朋友，我们只知道他每天参与了什么活动，而不知道他所在城市的天气是什么样的。
- 这个城市每天的天气情况，会和前一天的天气情况有点关系。譬如说，如果前一天是晴天，那么后一天是晴天的概率，就大于后一天是雨天的概率。
- 小明所在的城市，一年四季的天气情况都差不多。
- 小明每天会根据当天的天气情况，决定今天进行什么样的活动。
- 我们想通过小明的活动，猜测他所在城市的天气情况。

那么，城市天气情况和小明的活动选择，就构成了一个隐马尔科夫模型HMM

3.2、HMM隐马尔可夫模型

- HMM的基本定义：HMM是用于描述由隐藏的状态序列和显性的观测序列组合而成的双重随机过程。在前例中，城市天气就是隐藏的状态序列，这个序列是我们观测不到的。小明的活动就是观测序列，这个序列是我们能够观测到的。这两个序列都是随机序列。
- HMM的假设一：马尔可夫性假设。当前时刻的状态值，仅依赖于前一时刻的状态值，而不依赖于更早时刻的状态值。每天的天气情况，会和前一天的天气情况有点关系。
- HMM的假设二：齐次性假设。状态转移概率矩阵与时间无关。即所有时刻共享同一个状态转移矩阵。小明所在的城市，一年四季的天气情况都差不多。
- HMM的假设三：观测独立性假设。当前时刻的观察值，仅依赖于当前时刻的状态值。小明每天会根据当天的天气情况，决定今天进行什么样的活动。
- HMM的应用目的：通过可观测到的数据，预测不可观测到的数据。我们想通过小明的活动，猜测他所在城市的天气情况。



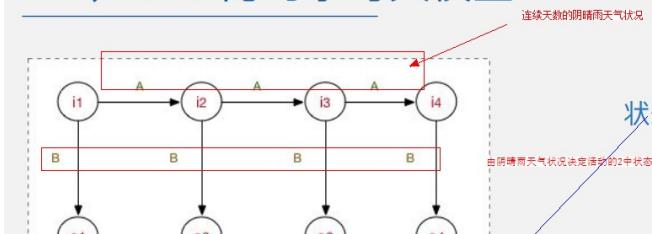
注一：马尔可夫性：随机过程中某事件的发生只取决于它的上一事件，是“无记忆”过程。

注二：HMM被广泛应用于标注任务。在标注任务中，状态值对应着标记，任务会给定观测序列，以预测其对应的标记序列。

他	和	爸爸	去	电影	院	看	哈利	波特
B-Per	O	B-Per	O	B-Loc	I-Loc	O	B-Per	I-Per

注三：HMM属于生成模型，是有向图。

3.2、HMM隐马尔可夫模型



状态转移概率矩阵A：

	晴天	阴天	雨天
晴天	0.5	0.2	0.3
阴天	0.3	0.5	0.2
雨天	0.2	0.3	0.5



3.2、HMM隐马尔可夫模型

约定一下HMM的标记符号，并通过套用上文的例子来理解：

- 状态值集合（一共有N种状态值）： (s_1, s_2, \dots, s_N)

天气的状态值集合为{晴天，阴天，雨天}。

- 观测值集合（一共有M种观测值）： (o_1, o_2, \dots, o_M)

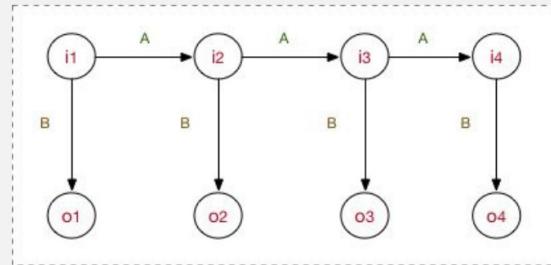
小明活动的观测值集合为{宅，打球}。

- 状态值序列： $y_1, y_2, \dots, y_t, \dots, y_T$

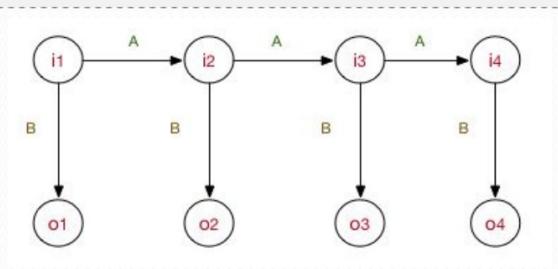
每一天的城市天气状态值构成的序列{晴晴晴阴雨晴}

- 观测值序列： $x_1, x_2, \dots, x_t, \dots, x_T$

每一天的小明活动的观测值构成的序列{球宅宅球宅宅}



3.2、HMM隐马尔可夫模型



- HMM模型的三个参数：A, B, π 。
- A：状态转移概率矩阵。表征转移概率，维度为N*N。
- B：观测概率矩阵。表征发射概率，维度为N*M。
- π ：初始状态概率向量。维度为N*1。
- $\lambda=(A, B, \pi)$ ，表示模型的所有参数。

状态转移概率矩阵A:

观测概率矩阵B:

初始概率状态向量 π :

3.2、HMM——概率计算问题

状态转移概率矩阵A:

	晴天	阴天	雨天
晴天	0.5	0.2	0.3
阴天	0.3	0.5	0.2
雨天	0.2	0.3	0.5

观测概率矩阵B:

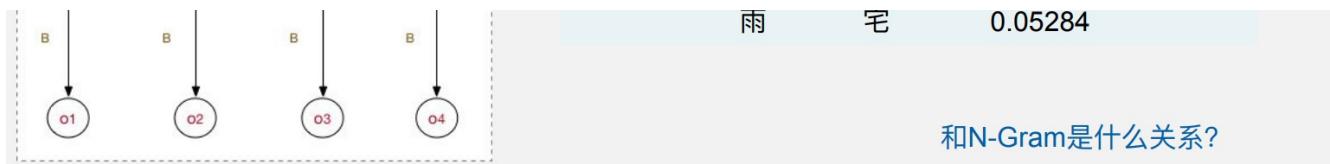
	宅	打球
晴天	0.5	0.5
阴天	0.4	0.6
雨天	0.7	0.3

初始概率状态向量 π :

晴天	0.2
阴天	0.4

简单算一下，已知模型参数，观测序列是{宅球宅}的概率

天数	天气	行为	概率	累计概率
第一天	晴	宅	0.10	0.54
	阴	宅	0.16	
	雨	宅	0.28	
第二天	晴	打球	0.077	0.248
	阴	打球	0.1104	
	雨	打球	0.0606	
第三天	晴	宅	0.04187	0.13022
	阴	宅	0.03551	



补充计算过程

很明显，在Graph图上， i_1, i_2, i_3 是指天气状况， O_1, O_2, O_3 是指观测状态。A矩阵是按列进行排列天数

由于第一天是不牵扯状态转移，是初始概率。因此比较容易写。

第一天：

$$\begin{aligned} P_{1\text{-}}(\text{晴-宅}) &= P_{\{\text{初始}\}}(\text{晴}) * P_B(\text{晴, 宅}) = 0.2 * 0.5 = 0.1 \\ P_{1\text{-}}(\text{阴-宅}) &= P_{\{\text{初始}\}}(\text{阴}) * P_B(\text{阴, 宅}) = 0.4 * 0.4 = 0.16 \\ P_{1\text{-}}(\text{雨-宅}) &= P_{\{\text{初始}\}}(\text{雨}) * P_B(\text{雨, 宅}) = 0.4 * 0.7 = 0.1 \end{aligned}$$

第二天需要考虑状态转移，所以需要使用状态转移概率矩阵A。并且第一天的状态当做是第二天的初始状态。

$$\begin{aligned} P_{2\text{-}}(\text{晴-打球}) &= P_B(\text{晴-打球}) * (P_{1\text{-}}(\text{晴-宅}) * P_A(\text{晴-晴}) + P_{1\text{-}}(\text{阴-宅}) * P_A(\text{晴-阴}) + P_{1\text{-}}(\text{雨-宅}) * P_A(\text{晴-雨})) = 0.5 * \\ (0.1 * 0.5 + 0.16 * 0.3 + 0.28 * 0.2) &= 0.077 \end{aligned}$$

$$P_{2\text{-}}(\text{阴-打球}) = P_B(\text{阴-打球}) * (P_{1\text{-}}(\text{晴-宅}) * P_A(\text{晴-阴}) + P_{1\text{-}}(\text{阴-宅}) * P_A(\text{阴-阴}) + P_{1\text{-}}(\text{雨-宅}) * P_A(\text{阴-雨})) = 0.6 * \\ (0.1 * 0.2 + 0.16 * 0.5 + 0.28 * 0.3) = 0.1104 \text{ }$$

$$P_{2\text{-}}(\text{雨-打球}) = P_B(\text{雨-打球}) * (P_{1\text{-}}(\text{晴-宅}) * P_A(\text{晴-雨}) + P_{1\text{-}}(\text{阴-宅}) * P_A(\text{阴-雨}) + P_{1\text{-}}(\text{雨-宅}) * P_A(\text{雨-雨})) = 0.3 * \\ (0.1 * 0.3 + 0.16 * 0.2 + 0.28 * 0.5) = 0.0606 \text{ }$$

第三天还是按照按照跟第二天相似的算法进行计算

- 以上我们采用的是前向计算累积概率的方法进行计算{宅球宅}的概率，此外我们还可以采用反向的方法来计算。

如何推算最有可能的天气状况？

3.2、(HMM——解码预测问题) && 维特比算法

状态转移概率矩阵A：

	晴天	阴天	雨天
晴天	0.5	0.2	0.3
阴天	0.3	0.5	0.2
雨天	0.2	0.3	0.5

观测概率矩阵B：

	宅	打球
晴天	0.5	0.5
阴天	0.4	0.6
雨天	0.7	0.3

初始概率状态向量π：

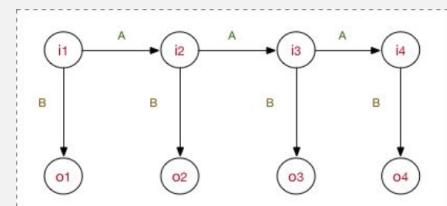
	晴天
阴天	0.4
雨天	0.4

已知模型参数，观测序列是{宅球宅}，最有可能的天气状况？

实际就是用动态规划求解概率最大路径。这时一条路径对应着一个状态序列。

根据动态规划原理，最优路径必须满足这样的特性：如果最优路径在时刻t通过节点 i_t^* 那么这一路径从 i_t^* 到终点 i_T^* 的部分路径，对于从 i_t^* 到 i_T^* 的所有可能来说，必须是最优的。因为假如不是这样，就会有一条新的最优路径。

所以只要递归的求在t时刻状态为i的各条路径的最大概率，直到时刻 $t=T$ ，此时最大的概率就是最优路径的概率P，同时也得到最优路径的终点。从这个终点逐步反推，即可得到最优路径，这就是维特比算法。



3.2、(HMM——解码预测问题) && 维特比算法

状态转移概率矩阵A：

	晴天	阴天	雨天
晴天	0.5	0.2	0.3
阴天	0.3	0.5	0.2
雨天	0.2	0.3	0.5

观测概率矩阵B：

已知模型参数，观测序列是{宅球宅}，最有可能的天气状况？

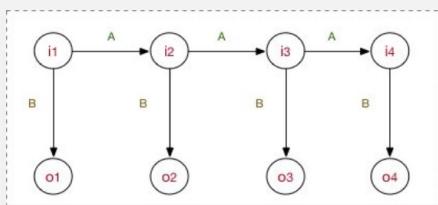
- 定义在时刻t状态为i的所有单个路径 (i_1, i_2, \dots, i_t) 中的概率最大值为：

$$\delta_t(s) = \max_{i_1, i_2, \dots, i_{t-1}} P(i_t = s, i_{t-1}, i_{t-2}, \dots, i_1, o_1, o_2, \dots, o_t | \lambda), s = 1, 2, \dots, N$$

	晴天	阴天
晴天	0.5	0.5
阴天	0.4	0.6
雨天	0.7	0.3

初始概率状态向量 π :

	晴天	0.2
阴天	0.4	
雨天	0.4	



那么 $t+1$ 的时刻，就是：

$$\delta_{t+1}(s) = \max_{i_1, i_2, \dots, i_t} P(i_{t+1} = s, i_t, i_{t-1}, i_{t-2}, \dots, i_1, o_1, o_2, \dots, o_t, o_{t+1} | \lambda), s = 1, 2, \dots, N$$

$$= \max_{1 \leq j \leq N} [\delta_t(j) a_{js}] b_i(o_{t+1})$$

再定义一个变量，用来回溯最大路径：在时刻 t 状态 i 的所有单个路径 $(i_1, i_2, \dots, i_{t-1}, i_t)$ 中，概率最大的路径第 $t-1$ 个节点为：

$$\varphi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], i = 1, 2, \dots, N$$

公式比较麻烦，但是还是可以理解记忆的。由于默认每个状态(例如{雨天-打球})是跟前一个的状态(天气)有关，但在进行预测的时候，只取最大的那个值。

$\$delta_{t+1} = \max_{1 \leq j \leq N} [\delta_t(j) \alpha_{js}] * b_i(O_{i+1})$

解释一下其中的参数， $\delta_t(j)$ 表示 t 时刻的概率状态变量， j 代表其矩阵中的某个索引值， α_{js} 代表的是状态转移概率矩阵 A ， $b_i(O_{i+1})$ 代表观测概率矩阵 B 。本质上还是利用了之前的HMM的概率计算公式，只不过是取最大值进行保留。

我们来看一下计算过程

3.2、(HMM——解码预测问题) && 维特比算法

状态转移概率矩阵 A :

	晴天	阴天	雨天
晴天	0.5	0.2	0.3
阴天	0.3	0.5	0.2
雨天	0.2	0.3	0.5

观测概率矩阵 B :

	宅	打球
晴天	0.5	0.5
阴天	0.4	0.6
雨天	0.7	0.3

初始概率状态向量 π :

	晴天	0.2
阴天	0.4	
雨天	0.4	

已知模型参数，观测序列是{宅球宅}，最有可能的天气状况？

维特比示例	第一天	第二天	第三天
	宅	打球	宅
雨天	0.28	0.042	0.0147
阴天	0.16	0.0504	0.01008
晴天	0.1	0.028	0.00756

对应到NLP情景中：

已知HMM模型参数——已知语料集

已知观测序列——“已知的句子”

最有可能的内部状态？——最有可能的词性序列？

竖着看，第一天

第一天在家里的天气情况三种（雨、阴、晴）

$$P_1\{\text{宅-雨}\}=0.4*0.7=0.28$$

$$P_1\{\text{宅-阴}\}=0.4*0.4=0.16$$

$$P_1\{\text{宅-雨}\}=0.2*0.5=0.1$$

关键就看第二天怎么算。

$$P_2\{\text{球-雨}\}=\max(P_1\{\text{宅-雨}\} * P_A\{\text{雨-雨}\} * P_B\{\text{球-雨}\})=0.28*0.5*0.3=0.042, P_1\{\text{宅-阴}\} * P_A\{\text{阴-雨}\} * P_B\{\text{球-阴}\}=0.16*0.2*0.3=0.016, P_1\{\text{宅-晴}\} * P_A\{\text{晴-雨}\} * P_B\{\text{球-晴}\}=0.1*0.3*0.3=0.009=0.042,$$

$$P_2\{\text{球-阴}\}=\max(P_1\{\text{宅-雨}\} * P_A\{\text{雨-阴}\} * P_B\{\text{球-阴}\})=0.28*0.3*0.6=0.0504, P_1\{\text{宅-阴}\} * P_A\{\text{阴-阴}\} * P_B\{\text{球-阴}\}=0.16*0.5*0.6=0.048, P_1\{\text{宅-晴}\} * P_A\{\text{晴-阴}\} * P_B\{\text{球-阴}\}=0.1*0.2*0.6=0.012=0.0504,$$

$$P_2\{\text{球-晴}\}=\max(P_1\{\text{宅-雨}\} * P_A\{\text{雨-晴}\} * P_B\{\text{球-晴}\})=0.28*0.2*0.5=0.028, P_1\{\text{宅-阴}\} * P_A\{\text{阴-晴}\} * P_B\{\text{球-晴}\}=0.16*0.3*0.5=0.024, P_1\{\text{宅-晴}\} * P_A\{\text{晴-晴}\} * P_B\{\text{球-晴}\}=0.1*0.5*0.5=0.025=0.028,$$

同理,按照球第二天的方法求第三天的概率,我们都求完后,进行反向回溯。

$$\varphi_t(i) = \arg \max_{1 \leq j \leq N} [\delta_{t-1}(j) a_{ji}], i = 1, 2, \dots, N$$

最终发现这个路径是最大的

维特比示例	第一天 宅	第二天 打球	第三天 宅
雨天	0.28	0.042	0.0147

即雨-雨-雨是最有可能的天气

- 到这里, HMM的方法基本就讲完了, 但怎么结合NLP去做中文NER, 需要找个例子说明下。

数据集采用的是HanLP提供的语料, 统计的状态集合是从人民日报中统计的次数

角色	意义	例子
A	上文	参与亚太经合组织的活动
B	下文	中央电视台报道
X	连接词	北京电视台和天津电视台
C	特征词的一般性前缀	北京电影学院
F	特征词的人名前缀	何镜堂纪念馆
G	特征词的地名性前缀	交通银行北京分行
K	特征词的机构名、品牌名前缀	中共中央顾问委员会 美国摩托罗拉公司
I	特征词的特殊性前缀	中央电视台 中海油集团
J	特征词的简称性前缀	巴政府
D	机构名的特征词	国务院侨务办公室
Z	非机构成分	
L	方位词	上游 东
M	数量词	36
P	数量+单位 (名词)	三维 两国
W	特殊符号, 如括号, 中括号	{ } []
S	开始标志	始###始

In [8]:

```
firpath = "D:/AI/交互文件夹/NER/HMM/data/"  
import os  
os.listdir(firpath)
```

Out[8]:

```
['emit_probability.txt',  
'initial_vector.txt',
```

```
'nt.pattern.txt',
'nt.tr.txt',
'nt.txt',
'transition_probability.txt']
```

In [38]:

```
with open(firldpath+'nt.txt', 'r', encoding="utf-8") as f:
    info1 = f.readlines()
```

In [39]:

```
info1
```

Out[39]:

```
['! A 3 B 2 D 1\n',
 '# A 1\n',
 '&rsquo; D 1\n',
 '' A 1\n',
 '( B 689 A 169 W 146 X 24\n',
 ') B 151 W 77 A 71\n',
 '* D 1\n',
 '+ A 2 W 1\n',
 ', A 299 B 45 W 7 X 2\n',
 '- A 50 W 27 X 14 B 8 D 1\n',
 '. A 71 W 11 B 1\n',
 '/ B 48 A 30 X 9\n',
 '1 D 6\n',
 '123 D 1\n',
 '13642667887 D 1\n',
 '16: B 1\n',
 '3: B 1\n',
 '4-3 D 1\n',
 '47: B 1\n',
 '4: B 1\n',
 '56079116 D 1\n',
 '88155377 D 1\n',
 ': B 62 A 18\n',
 '; A 21 B 5\n',
 '> D 2\n',
 '? W 61 A 12 D 4 B 1\n',
 '@ A 66\n',
 'A D 1\n',
 'CN D 1\n',
 'CO D 1\n',
 'Co D 1\n',
 'Company D 1\n',
 'FTTH-CHINA D 1\n',
 'GSC D 1\n',
 'LTD D 2\n',
 'Limited D 2\n',
 'Ltd D 2\n',
 '[ A 3 B 3 W 1\n',
 '\\ A 1\n',
 ] A 278 B 1 D 1\n',
 'co D 1\n',
 '- A 394 B 68 W 13 X 2\n',
 '- A 1\n',
 '' A 39 B 9\n',
 '' B 31 A 1\n',
 '' A 1729 B 700 X 7\n',
 '' A 1688 B 965 X 2\n',
 ... A 34 B 31\n',
 '→ A 10 B 6\n',
 '■ A 5\n',
 '□ A 5\n',
 '▲ A 5\n',
 '● A 13 B 1\n',
 '☆ W 1\n',
 ' D 2 W 2\n',
 '、 A 5677 X 3357 B 3178\n',
 '。 B 1966\n',
 '〈 A 12 B 1\n',
 '《 A 907 B 334 X 4\n',
 '》 B 101 A 17\n',
 '「 A 11\n',
 '【 A 29\n']
```

'】 A 78 B 3\n',
'(B 1\n',
'— P 254\n',
'— P 1\n',
'一举 B 2\n',
'一代 A 2\n',
'一公司 D 1\n',
'一共 B 7\n',
'一再 B 4\n',
'一再强调 B 1\n',
'一分 C 1\n',
'一分为二 B 1\n',
'一则 A 15\n',
'一卡通 A 1\n',
'一台 D 9\n',
'一同 B 3\n',
'一名 P 1\n',
'一向 B 3\n',
'一品 J 3\n',
'一团 P 1\n',
'一国两制 C 1\n',
'一场 D 18\n',
'一块 B 2 A 1\n',
'一大 B 1\n',
'一头 B 1\n',
'一如 P 1\n',
'一如既往 B 2\n',
'一姓 B 1\n',
'一定 B 2\n',
'一定会 B 1\n',
'一审 B 120\n',
'一家 A 132 B 19\n',
'一家之言 B 2\n',
'一家人 C 1\n',
'一局 D 24 P 8\n',
'一峰 P 1\n',
'一带 B 1\n',
'一年一度 B 2\n',
'一度 B 8\n',
'一心 P 1\n',
'一战 B 2\n',
'一把手 B 2\n',
'一方面 B 2 A 1\n',
'一旦 A 6 B 2\n',
'一期 B 15 A 1\n',
'一本 P 1\n',
'一条 P 1\n',
'一条龙 C 1\n',
'一枝独秀 B 1\n',
'一样 B 18 A 1\n',
'一格 P 2\n',
'一楼 B 12\n',
'一次性 B 1\n',
'一汽 C 7 B 1\n',
'一环 B 1\n',
'一百个 A 1\n',
'一直 B 68\n',
'一票否决权 B 1\n',
'一等 B 3\n',
'一米 P 1\n',
'一类 B 10\n',
'一系列 B 5\n',
'一级 J 9 B 4\n',
'一级方程式 A 2\n',
'一线 A 2 B 2\n',
'一组 D 14\n',
'一致 B 1\n',
'一致认为 A 1\n',
'一般 B 5 A 3\n',
'一行 B 7\n',
'一角 B 7\n',
'一贯 B 11\n',
'一费制 B 1\n',
'一起 B 15\n',
'一连 B 1\n',
'一道 B 4\n',
'一部 n 12\n'

'一间 P 2\n',
'一面 B 1\n',
'丁 J 6\n',
'七中 P 1\n',
'七仙岭 I 1\n',
'七匹狼 I 1\n',
'七大 C 4\n',
'七届 P 1\n',
'七彩 C 1\n',
'七彩虹 I 1\n',
'七星 I 41 A 1\n',
'七道 P 1\n',
'七里桥 I 1\n',
'万 P 132\n',
'万世 P 1\n',
'万事 C 1\n',
'万事如意 C 1\n',
'万事达 I 13\n',
'万事顺 I 1\n',
'万亿 P 5\n',
'万众 C 2\n',
'万佳 I 2\n',
'万全 P 2\n',
'万兴 I 1\n',
'有利 I 2\n',
'万叶 P 1\n',
'万向 I 1\n',
'万吨 B 13\n',
'万国 D 14 J 2\n',
'万宝 I 2\n',
'万家 P 3\n',
'万寿台 I 3\n',
'万寿桥 B 4\n',
'万户 P 1\n',
'万桥 P 1\n',
'万福 I 1\n',
'万福生 I 3\n',
'万科 I 37\n',
'万维 P 1\n',
'万缕 P 1\n',
'万达 I 1\n',
'万通 I 3\n',
'万通公司 D 1\n',
'万里 P 3\n',
'丈 P 1\n',
'三丰 I 3\n',
'三九 C 1\n',
'三令五申 B 1\n',
'三伏潭 I 1\n',
'三信 P 1\n',
'三元 J 9\n',
'三元桥 I 1\n',
'三公司 D 1\n',
'三军 C 4\n',
'三农 I 1\n',
'三分 C 2\n',
'三分线 B 1\n',
'三友 I 3\n',
'三发 P 3\n',
'三台 P 1\n',
'三合 P 2\n',
'三合村委会 D 1\n',
'三合顺 I 1\n',
'三和 I 4\n',
'三官乡 I 1\n',
'三山 I 3\n',
'三岔路口 L 1\n',
'三岛 P 1\n',
'三川 I 2\n',
'三希堂 I 1\n',
'三府 I 3\n',
'三得利 I 4\n',
'三战 B 2\n',
'三斗坪 I 1\n',
'三新 I 1\n',
'三星 K 386\n',
'三星级 T 2\n'

'一毛派' 1\n',
'三期 B 1\n',
'三木 C 1\n',
'三步踩 I 1\n',
'三水 P 2\n',
'三沙市 A 2\n',
'三洲 I 1\n',
'三田 I 1\n',
'三省 I 2\n',
'三级 J 16\n',
'三级士官 B 1\n',
'三组 D 1\n',
'三维 C 1 P 1\n',
'三联 I 2\n',
'三能 J 2\n',
'三菱 K 28\n',
'三角洲 C 1\n',
'三足鼎立 B 2\n',
'三轮车 C 1\n',
'三部 D 2\n',
'三里湾 I 1\n',
'三野 C 1\n',
'三鑫 I 5\n',
'三镇 P 1\n',
'三队 D 1\n',
'三马 P 1\n',
'三鸟 I 4\n',
'三鼎 I 2\n',
'上 L 119 B 101 A 65\n',
'上下 B 2 L 1\n',
'上世纪 B 1\n',
'上书 A 1\n',
'上交 A 5 B 1\n',
'上任 A 1\n',
'上半场 A 2\n',
'上台 B 1\n',
'上品折扣 A 6\n',
'上学 B 13\n',
'上官姓 I 2\n',
'上山 C 1\n',
'上市 C 40 B 6 A 2\n',
'上思 C 1\n',
'上报 A 53 B 8 X 2\n',
'上来 B 2\n',
'上林二路 A 1\n',
'上校 B 1\n',
'上汽 C 13\n',
'上浮利率 B 1\n',
'上海交大 K 14\n',
'上海分行 B 11\n',
'上海地区 A 1\n',
'上海浦 I 1\n',
'上海车展 A 1\n',
'上海队 K 22\n',
'上海青 I 1\n',
'上游 L 2\n',
'上演 B 12\n',
'上班 B 15\n',
'上班族 B 2\n',
'上级 A 3 B 1\n',
'上网 B 3 C 2\n',
'上网电价 B 4\n',
'上菱 I 1\n',
'上街 C 3 B 2\n',
'上访 B 9\n',
'上诉 C 4 B 3\n',
'上调 B 2\n',
'上述 A 4 B 2\n',
'下 B 16 A 15 L 14\n',
'下一代 A 1 C 1\n',
'下令 B 13 A 5\n',
'下关市 I 4\n',
'下午三点 B 2\n',
'下半场 A 1 B 1\n',
'下去 A 2\n',
'下发 B 120\n',
'下地 B 1\n',
'下基层 B 1\n'

'下塘边村 B 1\n',
'下士 B 2\n',
'下大力气 B 1\n',
'下属 B 61 X 5 A 2 D 1\n',
'下拨 B 8\n',
'下放 B 9 A 1\n',
'下文 B 4\n',
'下棋 B 1\n',
'下榻 A 4\n',
'下水 B 6\n',
'下洼 I 1\n',
'下浒山水库 A 2\n',
'下渡 I 1\n',
'下线 B 3\n',
'下西洋 B 1\n',
'下设 B 7\n',
'下调 B 2\n',
'下车 B 5\n',
'下载 B 1\n',
'下辖 B 11 A 1\n',
'下达 B 31\n',
'下面 B 8\n',
'不 B 63 A 19\n',
'不久 B 7 A 2\n',
'不乏 A 4 B 1\n',
'不仅 B 20 A 11\n',
'不会 B 18\n',
'不但 B 20 A 3\n',
'不具备 B 3\n',
'不再 B 10\n',
'不准 B 2\n',
'不利于 A 2\n',
'不动产 A 5 C 5\n',
'不及 A 3\n',
'不变 B 1\n',
'不可避免 B 2\n',
'不如 A 1\n',
'不妨 B 2\n',
'不宜 B 2\n',
'不属于 A 1\n',
'不干胶 C 1\n',
'不得 B 1\n',
'不得不 B 12\n',
'不想 B 2\n',
'不愧为 A 1\n',
'不愿 B 1\n',
'不懈 B 1\n',
'不拿 A 1 B 1\n',
'不敌 A 25\n',
'不断 B 25\n',
'不是 A 34\n',
'不服 A 3 B 1\n',
'不止 A 2\n',
'不正之风 C 4\n',
'不满 A 1 B 1\n',
'不理 C 13\n',
'不甘示弱 B 1\n',
'不用 B 6\n',
'不等 B 1\n',
'不约而同 B 1\n',
'不肯 B 1\n',
'不能 B 10 A 1\n',
'不要 B 7 A 2\n',
'不论是 A 2\n',
'不足 B 6\n',
'不过 A 24\n',
'不远处 B 1\n',
'不锈钢 C 23\n',
'不难 B 1\n',
'不顾 A 5 B 5\n',
'与 A 1143 B 318 X 212\n',
'与此同时 A 2\n',
'专 D 26 C 3 B 2\n',
'专业 C 133 B 7 A 6\n',
'专业人士 B 8\n',
'专业委员会 B 1\n',
'专业社论 D 1\n'

'专业权人 B 1\n',

'专业技术人员 B 1\n',

'专业运动员 B 7\n',

'专业音响 I 1\n',

'专供 B 1\n',

'专利 B 1 C 1\n',

'专利局 D 1\n',

'专升本考试 B 1\n',

'专卖 C 3\n',

'专卖局 D 19\n',

'专卖店 D 22 B 1\n',

'专员 C 6\n',

'专场 B 8\n',

'管委会 D 8\n',

'专家 B 138 C 17 A 5\n',

'专家学者 B 4\n',

'专家实地考察 B 1\n',

'专家局 D 4\n',

'专家建议 B 3\n',

'专家提醒 B 3\n',

'专家组 D 18\n',

'专攻 B 1\n',

'专柜 B 1 D 1\n',

'专案组 D 4\n',

'专注 B 1\n',

'专用 C 6 B 2\n',

'专用公路 B 1\n',

'专用汽车 I 1\n',

'专用设备 I 1\n',

'专电 B 11\n',

'专科 C 83 D 3\n',

'专科学校 D 22\n',

'专程 B 1\n',

'专稿 B 15\n',

'专线 D 2 B 1\n',

'专职 B 35 J 1\n',

'专职画家 B 1\n',

'专营店 D 7\n',

'专访 A 10 B 9\n',

'专辑 B 1\n',

'专门 B 47 P 17 A 1\n',

'专门从事 B 1\n',

'专项 B 5 J 2 A 1\n',

'专项基金 B 3\n',

'专题 B 11 C 7\n',

'专题报道 B 3\n',

'且 A 7\n',

'世 J 83 A 1\n',

'世亨 I 1\n',

'世博会 D 5 C 4\n',

'世嘉 K 2\n',

'世宗 I 2\n',

'世家 C 2\n',

'世源 I 1\n',

'世界 C 233 B 11 A 6\n',

'世界史 B 2\n',

'世界性 B 2\n',

'世界文化遗产 B 3\n',

'世界文化遗产名录 B 9\n',

'世界杯赛 A 1\n',

'世界经济 B 1\n',

'世界自然遗产名录 B 1\n',

'世界语 C 1\n',

'世界遗产 B 1\n',

'世界银行 K 33\n',

'世纪 C 29 A 2 D 1\n',

'世纪坛 I 6\n',

'世纪城 I 2\n',

'世茂 I 3\n',

'世贸 C 19\n',

'业 J 209 D 27 A 10 B 9\n',

'业主 A 1 C 1\n',

'业余 B 1 J 1\n',

'业务 B 11 C 8 D 3\n',

'业务员 B 1\n',

'业务科 B 5 D 1\n',

'业务部 D 62\n',

'业务部门 D 11\n',

'业务部 I 1\n',
'业务量 B 1\n',
'业绩 B 5\n',
'业绩快报 B 2\n',
'业绩考核 B 1\n',
'丛书 C 1\n',
'丛林 C 1\n',
'东 L 329 B 19 A 9\n',
'东丽 I 3\n',
'东五环 I 1\n',
'东京电力 A 4\n',
'东京队 K 1\n',
'东侧 B 5 A 1\n',
'东侨 I 1\n',
'东力 C 1\n',
'东北局 D 2 C 1\n',
'东北方向 B 4\n',
'东北民主联军 A 1\n',
'东北部 B 1 L 1\n',
'东升 C 10 D 3\n',
'东华 J 15\n',
'东南 L 40 A 1\n',
'东南快报 A 1\n',
'东周 C 1\n',
'东园 C 1 D 1\n',
'东国 I 5\n',
'东坝 I 15\n',
'东大 C 3\n',
'东大门 B 2 A 1\n',
'东宁 B 2\n',
'东宝 I 5\n',
'东富 I 2\n',
'东小口 I 1\n',
'东小营 I 1\n',
'东山加油站 D 1\n',
'东岗 I 3\n',
'东岩 I 1\n',
'东岭 I 6\n',
'东岳 C 2\n',
'东岳庙 A 1\n',
'东川 C 1\n',
'东府 I 1\n',
'东新城 I 14\n',
'东方 L 154\n',
'东方中乐团 B 1\n',
'东方学 I 2\n',
'东方红 I 8\n',
'东方网 I 2\n',
'东方路 I 3\n',
'东日 I 1\n',
'东旭 I 2\n',
'东昌 I 5\n',
'东明路 I 4\n',
'东星 I 3\n',
'东正教 I 2\n',
'东河 I 1\n',
'东泰 I 3\n',
'东泽 I 1\n',
'东洋 C 7\n',
'东流 C 1\n',
'东浩 I 1\n',
'东海岸 I 2 A 1\n',
'东湖路 A 1\n',
'东瓯 I 1\n',
'东盛 I 3\n',
'东移 B 1\n',
'东站 D 92\n',
'东线 A 1 L 1\n',
'东翼 B 4\n',
'东联 I 1\n',
'东航 C 8 A 1\n',
'东芝 K 14\n',
'东苑 B 1\n',
'东莞公司 D 1\n',
'东莞队 K 22\n',
'东街口 I 6\n',
'东西 C 6\n',
'.\n'.

'东四门 C 1\n',
'西湖 I 5\n',
'东路 C 9 A 1 B 1\n',
'东辉 I 1\n',
'东道主 A 1\n',
'东邦 I 1\n',
'东郊 L 9 A 2\n',
'东部 L 5\n',
'东门 B 13 C 4\n',
'东门外 B 1\n',
'东阿 C 2\n',
'东顺 I 1\n',
'东风 C 68 A 2 B 2\n',
'东风桥 I 1\n',
'丝 C 6\n',
'丝丽雅 I 2\n',
'丝印 I 2\n',
'丝宝 I 1\n',
'丝弦 C 1\n',
'丝绸 C 7\n',
'丝绸之路 C 2\n',
'丝网 C 2\n',
'丝网厂 D 1\n',
'丝网花 I 1\n',
'丢 A 1\n',
'两 P 7\n',
'两会 D 195 C 7 A 1\n',
'两党 D 1\n',
'两分 B 1\n',
'两国 P 7\n',
'两地 B 2\n',
'两家 P 1\n',
'两岸 C 46 A 1\n',
'两性生活 B 1\n',
'两柄 B 15\n',
'两江 I 1\n',
'两淮 I 2 B 1\n',
'两省 A 8 I 2\n',
'严厉 B 5\n',
'严厉打击 B 1\n',
'严密 B 2\n',
'严岛 I 1\n',
'严明 B 3\n',
'严查 B 3 C 1\n',
'严格 B 19\n',
'严格执行 A 21\n',
'严格控制 A 1\n',
'严格遵守 B 3 A 1\n',
'严正 B 1\n',
'严禁 B 1\n',
'严管 I 7\n',
'严肃 B 5\n',
'严肃查处 B 3 A 2\n',
'丧事 C 1\n',
'丧失 B 1\n',
'个 A 12 B 4\n',
'个人 A 9 B 5\n',
'个人信息 B 1\n',
'个体 C 2 B 1\n',
'个别 B 1\n',
'个性化 B 1\n',
'个私 J 1\n',
'个贷 C 1\n',
'丫丫 I 1\n',
'中 L 1899 B 120 A 35 X 1\n',
'中世纪 C 1\n',
'中东和平进程 A 1\n',
'中亚国家 A 1\n',
'中介 A 4 C 2\n',
'中企 J 1\n',
'中侨 I 1\n',
'中信 I 2\n',
'中信银行信用卡 A 6\n',
'中储粮 I 3\n',
'中共 A 460 C 426\n',
'中共中央 K 724\n',
'中兴 C 39\n',
.....

'中农资源 I 1\n',
'中冶 J 1\n',
'中凯 I 1\n',
'中化 J 2\n',
'中医 C 69 A 5 B 1\n',
'中医学 B 1\n',
'中医科 D 6\n',
'中医药 C 63 A 3\n',
'中医院 D 131\n',
'中华 I 102 A 7\n',
'中华全国总工会 K 10\n',
'中华民国 C 3 A 1\n',
'中华玉 I 1\n',
'中华苏维埃共和国 A 12\n',
'中华鲟 P 4\n',
'中印边界问题 A 2\n',
'中和 C 10 A 1\n',
'中咨 J 2\n',
'中国与世界经济 B 3\n',
'中国之声 B 2\n',
'中国人 A 1\n',
'中国人民政治协商会议 A 1\n',
'中国人民解放军 K 82\n',
'中国人民解放军海军航空兵 A 3\n',
'中国人民银行 K 91\n',
'中国公共财政 B 1\n',
'中国共产党 K 503\n',
'中国出版政府奖 A 1\n',
'中国县域经济报 B 1\n',
'中国台湾 A 3\n',
'中国台湾网 A 5\n',
'中国建材 A 1 I 1\n',
'中国摄影报 A 1\n',
'中国文化博览 B 2\n',
'中国文化研究 B 3\n',
'中国文学研究 B 1\n',
'中国日报 A 1 B 1\n',
'中国星 K 13\n',
'中国服装 A 1\n',
'中国画 C 1\n',
'中国留学生 A 1\n',
'中国科学院 K 222\n',
'中国移动 A 3 I 1\n',
'中国经济 A 1\n',
'中国绘画 B 1\n',
'中国网 I 6\n',
'中国网事 B 1\n',
'中国联通公司 D 1\n',
'中国航空 A 1\n',
'中国西部 B 1\n',
'中国邮政 A 1\n',
'中国铁建 A 1\n',
'中国银行 K 175\n',
'中国银行业 B 6\n',
'中国队 K 203\n',
'中国非物质文化遗产 B 2\n',
'中国馆 C 2\n',
'中场 B 5\n',
'中坚 C 1\n',
'中外 J 3\n',
'中外合资 C 1\n',
'中外运 I 1\n',
'中大 I 2\n',
'中天 I 3\n',
'中天新闻 A 4\n',
'中央 C 3499 B 33 D 26 A 21\n',
'中央企业 A 1\n',
'中央军委 K 376\n',
'中央委员 B 5\n',
'中央局 C 2\n',
'中央政府 D 11\n',
'中央苏区 A 6\n',
'中央银行 K 28\n',
'中孚 I 1\n',
'中学 D 622\n',
'中实 C 1\n',
'中宣部 C 5\n',

'中将 B 1\n',
'中小企业 C 37 A 1\n',
'中小学 C 2 B 1\n',
'中小学校 D 3\n',
'中山大学 K 34\n',
'中山街 I 1\n',
'中常委 D 3\n',
'中广 I 1\n',
'中广核 I 5\n',
'中广网 B 1\n',
'中建 J 2\n',
'中影 C 6\n',
'中心 D 4570 A 2 B 2\n',
'中心城区 A 1\n',
'中心店 D 2\n',
'中心校 D 12\n',
'中心站 D 50\n',
'中心组 D 6\n',
'中恒 I 1\n',
'中成药 B 1\n',
'中投顾问 A 1\n',
'中文 I 14 B 3\n',
'中文系 D 15\n',
'中文网 B 2\n',
'中新社 K 403\n',
'中方 A 4 B 1\n',
'中标 B 4\n',
'中校 B 2\n',
'中毒 C 1\n',
'中油 C 3\n',
'中泰 I 1\n',
'中海国际社区 B 2\n',
'中海油 I 4\n',
'中港 I 4\n',
'中游 C 3\n',
'中澳 I 1\n',
'中爪哇 I 1\n',
'中环 L 25\n',
'中电投 I 9\n',
'中电联 K 1\n',
'中直机关 C 1\n',
'中石化 I 57 A 6\n',
'中石油 I 27 A 13\n',
'中科 J 3\n',
'中科大 I 7\n',
'中科院 C 144 A 42\n',
'中等 J 4\n',
'中粮 I 11\n',
'中级 J 590\n',
'中纪委 K 477\n',
'中纺 C 1\n',
'中组部 A 2\n',
'中经网 I 1\n',
'中美 C 36\n',
'中美史克 B 8\n',
'中联 I 2\n',
'中航 J 2\n',
'中航技 J 1\n',
'中船 C 1\n',
'中药 C 8 B 5\n',
'中药房 B 2 D 1\n',
'中药材 C 2\n',
'中西 C 1\n',
'中西医 C 21\n',
'中西比较 B 1\n',
'中试 C 4\n',
'中资 C 6\n',
'中超 A 4\n',
'中路 C 4 B 1\n',
'中转 B 2\n',
'中远 I 1\n',
'中通速递 A 3\n',
'中邮 I 1\n',
'中部 A 1 L 1\n',
'中部非洲 A 1\n',
'中铁 K 58\n',
'中队 D 161\n',
.....

'中队长 B 21\n',
'中青年 C 6 B 2\n',
'中青旅 I 1\n',
'中革军委 K 39\n',
'中风 B 1\n',
'中餐厅 D 1\n',
'中餐馆 D 1\n',
'丰 P 64\n',
'丰乐 I 5\n',
'丰华 I 1\n',
'丰原 I 1\n',
'丰富 A 1\n',
'丰富多彩 C 1\n',
'丰庆路 I 3\n',
'丰收 B 2 C 1\n',
'丰桥 B 4 I 1\n',
'丰泰 I 2\n',
'丰源 I 4\n',
'丰特 I 1\n',
'丰田 I 23 A 2\n',
'丰盛 C 1\n',
'丰美 C 1\n',
'丰达 C 1\n',
'临 C 6 A 3\n',
'临客 B 1\n',
'临客列车 B 3\n',
'临平 I 1\n',
'临床 C 22 B 5 A 1\n',
'临床医学 B 1\n',
'临床医学博士 B 1\n',
'临床检验 B 1\n',
'临床肿瘤学 B 1\n',
'临床营养 B 1\n',
'临时 B 23 J 18 A 7\n',
'临时代办 B 7\n',
'临时聘用人员 B 1\n',
'临河 C 1\n',
'临河里 I 2\n',
'临泉 I 13\n',
'临海 C 8\n',
'临涣 I 9\n',
'临清 C 1\n',
'临港 I 4\n',
'临澜湾 B 1\n',
'临终 C 8\n',
'临街 B 2\n',
'\` A 1\n',
'丸善 I 1\n',
'丹 J 4\n',
'丹丹 A 2\n',
'丹桂 B 7\n',
'丹霞 I 3\n',
'为 A 1050 B 322 P 31 X 4\n',
'为主 B 3\n',
'为了 B 33 A 1\n',
'为什么 A 11 B 9\n',
'为什么要来 A 3\n',
'为何 A 5 B 5\n',
'为基础 B 1\n',
'为数不多 B 1\n',
'为数不少 B 1\n',
'为期 B 4\n',
'为此 B 17 A 3\n',
'为民 B 3\n',
'为首 B 2\n',
'主 B 4\n',
'主人 B 1\n',
'主任 B 1339\n',
'主任医师 B 54\n',
'主任委员 B 17\n',
'主任科员 B 7\n',
'主会场 X 3\n',
'主体 B 6\n',
'主力 B 10\n',
'主办 B 147 A 1\n',
'主办单位 A 4\n',
'主办方 A 5\n',

'主动 B 10\n',
'主唱 B 2\n',
'主场 B 26\n',
'主城区 B 2\n',
'主委 B 51\n',
'主管 B 3\n',
'主导 B 4\n',
'主将 B 2\n',
'主帅 B 26\n',
'主席 B 709 X 7 A 5\n',
'主席团 D 106\n',
'主张 A 8 B 3\n',
'主承销商 A 6\n',
'主持 B 29 A 23\n',
'主持人 B 10\n',
'主政 A 1\n',
'主教练 B 30\n',
'主权财富基金 A 1\n',
'主板 B 3\n',
'主治 B 2\n',
'主治医师 B 15\n',
'主治医生 B 4\n',
'主流 A 2 C 1\n',
'主管 B 48 A 4\n',
'主管单位 A 5\n',
'主管部门 B 8\n',
'主编 B 2\n',
'主要 B 30\n',
'主要负责人 B 33\n',
'主要领导 B 16\n',
'主计 I 1\n',
'主页 B 1\n',
'主题 C 5\n',
'主题乐园 B 1\n',
'丽 P 20 A 2\n',
'丽人 C 5\n',
'丽华 I 3\n',
'丽园 I 1\n',
'丽思 I 3\n',
'丽新 I 2\n',
'丽日 C 1\n',
'丽景 I 1\n',
'丽晶 I 2\n',
'丽江火车站 B 3\n',
'丽泽 I 1\n',
'丽都 I 8\n',
'丽雅 I 2\n',
'举 A 1\n',
'举办 B 95 A 8\n',
'举报 A 69 B 32 C 4 X 1\n',
'举行 B 427 A 2\n',
'举行仪式 B 1\n',
'举起 A 1\n',
'举重队 D 2\n',
'乃 A 4\n',
'乃至 A 3 B 1\n',
'久 C 4\n',
'久久 P 2\n',
'久保 C 1\n',
'久安 I 1\n',
'久尔 A 1\n',
'久远 C 3\n',
'久隆 C 2\n',
'义 J 17 B 5\n',
'义务 C 5 B 1\n',
'义工 C 27\n',
'之 B 16 A 1\n',
'之一 B 15 A 2\n',
'之上 B 4\n',
'之中 B 1\n',
'之前 B 19 A 8\n',
'之后 B 22 A 3\n',
'之外 B 3\n',
'之家 P 1\n',
'之意 B 1\n',
'之所以 B 9\n',
'之旅 I 6\n',

'之间 B 45\n',
'之际 A 1\n',
'乌 J 62 A 5\n',
'乌什 B 1\n',
'乌兰察 I 3\n',
'乌奎高速公路 A 13\n',
'乌市 J 1\n',
'乌海市乌达 A 7\n',
'乌石 I 1\n',
'乌素 C 1\n',
'乌金 C 2\n',
'乌鲁木齐火车站 B 2\n',
'乌龙指 B 1\n',
'乐 C 34 A 1\n',
'乐业 C 1\n',
'乐东 I 2\n',
'乐事 C 1\n',
'乐凯 I 1\n',
'乐华 I 1\n',
'乐善 I 1\n',
'乐器 C 5\n',
'乐器厂 D 2\n',
'乐团 D 73\n',
'乐园 B 3 C 1\n',
'乐天 C 13\n',
'乐施会 D 12\n',
'乐欢 I 1\n',
'乐语 A 2\n',
'乐购 I 2\n',
'乐透 I 6\n',
'乐道 I 8\n',
'乐邦 I 1\n',
'乐队 D 83\n',
'乐风 I 1\n',
'乐高 K 8\n',
'乒 P 2\n',
'乒乓 C 17 B 1\n',
'乒乓球 C 13\n',
'乒协 D 5\n',
'乒羽 I 4\n',
'乔 P 1\n',
'乘 B 4 C 4 A 1\n',
'乘务员 C 1\n',
'乘坐 B 9 A 5\n',
'乘火车 B 1\n',
'乘胜 I 1\n',
'乘警 B 7 C 4\n',
'乘车 B 10 A 1\n',
'乙炔 C 1\n',
'乙等 I 1\n',
'乙肝疫苗 B 2\n',
'九 J 19\n',
'九三学社 K 56\n',
'九华山 I 9\n',
'九华山风景区 X 1\n',
'九堡 I 1\n',
'九天 C 1\n',
'九家 A 3 I 1\n',
'九峰 I 6\n',
'九方 I 2\n',
'九日 C 1\n',
'九歌 I 1\n',
'九毛九 A 1\n',
'九洲 I 40\n',
'九源 I 1\n',
'九溪 I 1\n',
'九牧王 I 2\n',
'九环线 A 1\n',
'九老 I 5\n',
'九色鹿 I 1\n',
'九运会 C 1\n',
'九通 P 1\n',
'九鼎 C 5\n',
'九龙仓 K 6\n',
'九龙坡 I 5\n',
'九龙城 A 1\n',
'九龙塘 A 2\n',

```
'九龙湾 I 1\n',
'也 B 509 A 1\n',
'也许 A 1\n',
'乡 C 39 A 12 B 1\n',
'乡下 L 2\n',
'乡党委 D 7\n',
'乡土 C 1\n',
'乡政府 D 4\n',
'乡村 A 1 B 1 C 1\n',
'乡村游 I 1\n',
'乡级 A 1\n',
'乡镇 C 22 A 8 B 2\n',
'乡镇企业 C 2\n',
'乡长 B 1\n',
'书 B 1 C 1\n',
...]
```

- 拿第一行来讲，!这个符号作为上文出现了3次，作为下文出现了2次，作为机构的特征词出现了1次。其他未出现的状态可以初始化为0次

In [40]:

```
with open(firlpath+'nt.tr.txt', 'r', encoding="utf-8") as f1:
    info2 = f1.readlines()
```

In [41]:

```
info2
```

Out[41]:

```
[',A,B,C,D,F,G,I,J,K,L,M,P,S,W,X,Z\n',
'A,0,0,19945,883,2013,58781,3290,1582,19254,1422,282,944,0,0,0,0,0\n',
'B,3013,0,0,0,0,0,0,0,0,0,0,0,0,125708\n',
'C,0,0,25949,57230,142,1908,850,1603,53,169,260,834,0,144,0,0\n',
'D,0,109511,4389,6473,135,1018,476,229,28,105,177,120,0,59,4586,0\n',
'F,0,0,971,2666,138,127,92,163,5,7,5,87,0,48,0,0\n',
'G,0,0,24497,42962,1283,5178,3104,2782,182,1415,756,1173,0,140,0,0\n',
'I,0,0,2515,5556,34,270,179,181,3,39,56,210,0,11,0,0\n',
'J,0,0,2002,4973,69,96,85,707,4,95,47,535,0,23,0,0\n',
'K,0,19920,1162,2036,13,184,64,57,16,3,32,25,0,0,1238,0\n',
'L,0,0,1289,1476,19,68,58,481,1,18,10,289,0,0,0,0\n',
'M,1000,0,569,758,1,7,11,128,0,76,86,143,0,1,0,0\n',
'P,0,0,1647,1989,70,54,200,425,0,67,29,433,0,9,0,0\n',
'S,9509,0,2911,104,278,12704,476,212,4031,210,23,86,0,0,0,1131650\n',
'W,0,0,123,150,23,105,11,9,0,4,0,10,0,0,0,0\n',
'X,0,0,1173,50,91,2972,158,77,1173,79,17,34,0,0,0,0\n',
'Z,95874,0,0,0,0,0,0,0,0,0,0,0,0,0,19796133\n']
```

上述文件内容是标签转移矩阵，第一行和第一列是表头，中间代表的数字的含义：

0 --- 站位，表示没有标签转移关系；

非零 --- 表示该行标签转向列标签的转移次数。比如第二行第四列 19945代表 A标签之后接着C标签这种情况出现了19945次

讲到这里，我们来看下前辈们写过的代码，生成观测概率矩阵和状态转移概率矩阵

In []:

In [42]:

```
## 生成初始状态概率矩阵
### 其实很简单的公式，就是隐藏状态标签的次数和概率
def generate_initial_vector(hidden_states):
    """
    生成初始化概率向量，命名为initial_vector.txt，格式每一行为：状态,出现次数,概率
    :param hidden_states: 隐藏状态list
    :return:
    """
    the_hidden_states = {x:0 for x in hidden_states}
    count = 0 # 计算总数
```

```

with open("./data/nt.txt",mode='r') as nrfile:
    all_data = nrfile.readlines()
    for line in all_data:
        tags_and_freq = line.strip().split(" ")[1:]
        for index in range(0,len(tags_and_freq),2):
            tmp_list = tags_and_freq[index:index+2] #list的第一个元素为状态标识，第二个元素为数量
            the_hidden_states[tmp_list[0]] += eval(tmp_list[1])
            count += eval(tmp_list[1])
with open("./data/initial_vector.txt",mode="w") as outputfile:
    for key,value in the_hidden_states.items():
        str_to_write = "%s,%d,%f\n" %(key,value,float(value)/count)
        outputfile.write(str_to_write)
print ("generated ./data/initial_vector.txt")

```

生成的文件内容

In [44]:

```
%mark
A,109354,0.060114
B,137127,0.075381
C,101101,0.055577
D,135523,0.074499
F,6781,0.003728
G,92134,0.050647
I,11794,0.006483
J,10395,0.005714
K,24924,0.013701
L,3962,0.002178
M,1985,0.001091
P,6305,0.003466
S,1169907,0.643116
W,895,0.000492
X,6931,0.0003810
Z,5,0.000003
```

UsageError: Line magic function `%mark` not found.

In [45]:

```
### 生成状态转移矩阵
def generate_transition_probability(hidden_states):
    """
    生成转移概率矩阵，命名为transition_probability.txt; 格式为每一行：状态1,状态2,概率
    :param hidden_states: 隐状态
    :return:
    """
    initial_count = {x: 0 for x in hidden_states} # 初始化计数
    result = []
    with open("./data/nt.tr.txt",mode="r") as initial_count_file:
        all_data = initial_count_file.readlines()
        for line in all_data[1:]:
            split_line = line.strip().split(",")
            first_state = split_line[0]
            the_sum = sum([eval(number) for number in split_line[1:]])
            for index,second_state in enumerate(hidden_states):
                result.append([first_state,second_state,str(float(split_line[1:][index])/the_sum)])
    # 输出、写入文件
    with open("./data/transition_probability.txt",mode="w") as output_file:
        for thelist in result:
            str_to_write = "%s,%s,%s\n" %(thelist[0],thelist[1],thelist[2],)
            output_file.write(str_to_write)
    print ("generated ./data/transition_probability.txt")
```

- 生成的文件

In [51]:

```
%mark
A,A,0.0
A,B,0.0
A,C,0.1840012546588435
A,D,0.008146057050075648
A,F,0.018570795970331008
```

```
..., ..., ..., ..., ..., ...
A,G,0.5422801579394073
A,I,0.030351673493486844
A,J,0.014594634488357504
A,K,0.17762648068194398
A,L,0.013118565260710728
A,M,0.002601572013727444
A,P,0.008708808443115982
A,S,0.0
A,W,0.0
A,X,0.0
A,Z,0.0
B,A,0.023407214052097173
B,B,0.0
B,C,0.0
B,D,0.0
B,F,0.0
B,G,0.0
B,I,0.0
B,J,0.0
B,K,0.0
B,L,0.0
B,M,0.0
B,P,0.0
B,S,0.0
B,W,0.0
B,X,0.0
B,Z,0.9765927859479028
```

```
UsageError: Line magic function `%mark` not found.
```

In [47]:

```
### 获得发射矩阵，也就是观测概率矩阵
def generate_emit_probability(initial_freq):
    """
    生成发射矩阵，命名为emit_probability.txt；格式为每一行：隐状态,显状态,概率
    :param initial_freq: 隐状态初始化出现频数，是一个字典，key为隐状态标识，value为频数
    :return:
    """
    result = []
    with open("./data/nt.txt", mode="r") as nrfile:
        all_data = nrfile.readlines()
        for line in all_data:
            split_line = line.strip().split(" ")
            observed_state = split_line[0]
            tags_and_freq = split_line[1:]
            for index in range(0, len(tags_and_freq), 2):
                tmp_list = tags_and_freq[index:index+2] #list的第一个元素为隐状态标识，第二个元素为数量
                result.append([tmp_list[0], observed_state, float(tmp_list[1])/initial_freq[tmp_list[0]]])
    # 输出、写入文件
    with open("./data/emit_probability.txt", mode="w") as output_file:
        for thelist in result:
            str_to_write = "%s,%s,%s\n" % (thelist[0], thelist[1], str(thelist[2]))
            output_file.write(str_to_write)
    print ("generated ./data/emit_probability.txt")
```

In [48]:

```
def get_initial_freq():
    """
    获取每个标签出现的频数
    :return: 字典，key为标签，value为频数
    """
    result = {}
    with open("./data/initial_vector.txt", mode="r") as file:
        all_data = file.readlines()
        for line in all_data:
            split_line = line.strip().split(",")
            if len(split_line) == 3:
                result[split_line[0]] = int(split_line[1])
    return result
```

In []:

```

if __name__ == '__main__':
    hidden_states = ["A", "B", "C", "D", "F", "G", "I", "J", "K", "L", "M", "P", "S", "W", "X", "Z"]
    generate_initial_vector(hidden_states)
    generate_transition_probability(hidden_states)
    generate_emit_probability(get_initial_freq())

```

接下来我们看下怎么通过一个已知观测序列，怎么求隐藏状态

In []:

```

# -*- coding:utf-8 -*-
# -*- coding:utf-8 -*-
# 作者: 李鹏飞
# 个人博客: https://www.lookfor404.com/
# 代码说明: https://www.lookfor404.com/用隐马尔可夫模型hmm做命名实体识别-ner系列二/
# github项目: https://github.com/lipengfei-558/hmm_ner_organization
import jieba
class OrgRecognize:
    def __init__(self, input_sentence):
        self.hidden_states = ["A", "B", "C", "D", "F", "G", "I", "J", "K", "L", "M", "P", "S", "W", "X", "Z"]
        self.observed_states = self.get_observed_states(sentence=input_sentence)
        self.initial_vector = self.load_initial_vector()
        self.transition_matrix = self.load_transition_matrix(hidden_states=self.hidden_states)
        self.emission_matrix = self.load_emission_matrix(hidden_states=self.hidden_states)

    def load_patterns(self):
        """
        读取机构名模式串
        :return: list, 元素为模式串
        """
        result = []
        with open("./data/nt.pattern.txt", "r", encoding='utf-8') as file:
            datas = file.readlines()
            for line in datas:
                result.append(line.strip())
        return result

    def load_transition_matrix(self, hidden_states):
        """
        载入状态转移矩阵
        :return: 字典: key为首状态, value为字典--key为次状态, value为概率
        """
        result = {x: {} for x in hidden_states}
        with open("./data/transition_probability.txt", "r", encoding='utf-8') as file:
            datas = file.readlines()
            for line in datas:
                split_line = line.strip().split(",")
                result[split_line[0]][split_line[1]] = split_line[2]
        return result

    def load_initial_vector(self):
        """
        载入初始化向量
        :return: 字典: key为隐状态标识, value为概率
        """
        result = {}
        with open("./data/initial_vector.txt", "r", encoding='utf-8') as file:
            datas = file.readlines()
            for line in datas:
                split_line = line.strip().split(",")
                result[split_line[0]] = split_line[2]
        return result

    def load_emission_matrix(self, hidden_states):
        """
        载入发射矩阵
        :param hidden_states: 隐藏状态list
        :return: 字典, 格式为: key为隐状态, value是一个字典--key为观察状态, value为概率
        """
        result = {x: {} for x in hidden_states}
        with open("./data/emit_probability.txt", "r", encoding='utf-8') as file:
            datas = file.readlines()
            for line in datas:
                split_line = line.strip().split(" ")

```

```

        split_line = line.split(').split(' ')
        result[split_line[0]][split_line[1]] = split_line[2]
    return result

def get_observed_states(self, sentence):
    return sentence

def viterbi(self, observation, hidden_states, initial_probability, transition_probability, emit_probability):
    """
    用维特比算法计算最优标签
    :param observation: 粗分词结果
    :param hidden_states: 隐藏状态标签, 最终要求的标签都在里面
    :param initial_probability: 初始状态矩阵
    :param transition_probability: 转移状态矩阵
    :param emit_probability: 发射矩阵
    :return: 最优标签
    """
    result = []
    compute_recode = [] # 记录每一次的计算结果
    # 初始化
    tmp_result = {}
    for state in hidden_states:
        if observation[0] in emit_probability[state]:
            tmp_result[state] = eval(initial_probability[state]) * eval(emit_probability[state][observation[0]])
        else:
            tmp_result[state] = 0
    compute_recode.append(tmp_result)

    # 对于之后的词语, 继续计算
    for index, word in enumerate(observation[1:]):
        tmp_result = {}
        for current_state in hidden_states:
            # 取最大值: 上一次的所有状态(x) * 转移到当前状态 (current_state) * 发射概率
            if word in emit_probability[current_state]:
                tmp_result[current_state] = max(
                    [compute_recode[index][x] * eval(transition_probability[x][current_state]) * eval(emit_probability[current_state][word]) for x in hidden_states])
            else:
                tmp_result[current_state] = 0
        compute_recode.append(tmp_result)

    # 返回概率最大的标签序列
    tag_sequence = []
    for recode in compute_recode:
        tag_sequence.append(max(recode, key=recode.get))
    return tag_sequence

def get_organization(self, observation, sequence, patterns):
    """
    得到识别的机构名
    :param observation: 单词序列
    :param sequence: 标注序列
    :param patterns: 模式串
    :return: list, 机构名
    """
    org_indices = [] # 存放机构名的索引
    orgs = [] # 存放机构名字符串
    tag_sequence_str = ''.join(tag_sequence) # 转为字符串
    for pattern in patterns:
        if pattern in tag_sequence_str:
            start_index = (tag_sequence_str.index(pattern))
            end_index = start_index + len(pattern)
            org_indices.append([start_index, end_index])
    if len(org_indices) != 0:
        for start, end in org_indices:
            orgs.append(''.join(observation[start:end]))
    return orgs

if __name__ == '__main__':
    sentence = ["始##始", "中海油", "集团", "在", "哪里", "末##末"]
    orgrecog = OrgRecognize(sentence)
    observation = sentence
    initial_probability = orgrecog.load_initial_vector()

```

```
transition_probability = orgrecog.load_transition_matrix(hidden_states=orgrecog.hidden_states)
emit_probability = orgrecog.load_emission_matrix(hidden_states=orgrecog.hidden_states)
tag_sequence = orgrecog.viterbi(observation=observation, hidden_states=orgrecog.hidden_states,
                                 initial_probability=initial_probability,
                                 transition_probability=transition_probability, emit_probability=emit_probability)
print(tag_sequence)
patterns = orgrecog.load_patterns()
results = orgrecog.get_organization(observation=observation, sequence=tag_sequence, patterns=patterns)
if len(results) == 0:
    print("未识别到机构名")
    print(tag_sequence)
else:
    for result in results:
        print(result)
```

In []:

篇末读取多个excel并进行合并的技巧

In [1]:

```
import openpyxl
import pandas as pd
import os
```

In [5]:

```
filepath = "D:/mobvista/计费结算/2020年1月份消费结算/香港2/"
names = os.listdir(filepath)
df = locals()
for i in range(len(names)):
    df["df"+str(i)] = pd.DataFrame(pd.read_csv(filepath+names[i]))
```

In [6]:

```
result = pd.concat([df0, df1, df2, df3, df4], sort = False)
```

In [7]:

```
writer = pd.ExcelWriter(filepath+'香港2.xlsx')
result.to_excel(writer, index=False)
writer.save()
```

In [54]:

```
3607.63+3627.33+159.46+2765.82
```

Out[54]:

```
10160.24
```

In [55]:

```
1937.58+1278.69+2034.99+2433.35+2286.22
```

Out[55]:

```
9970.83
```

In [56]:

```
71747.089+14825.38
```

Out[56]:

```
86572.46900000001
```

In [57]:

```
8550.225571 +1564.02
```

Out[57]:

```
10114 245571000001
```

In [58]:

```
15021.39771+2286.62
```

Out[58]:

```
17308.01771
```

In []: