1. 观察集 inputfile.utf8.
已经分词好的文本
eg.
今天 是 星期一

2° 文本标记
1° 按空格对每行字符串分割!
2° 按字节长度和字符串的长度进行标记.
eg. 今天 今/B天/E
6个字节
substr(0,3)+'B'
+substr(3,3)+'E'
*这里只考虑了常用汉字
采用 utf8 编码. 一个汉字占3个字节
欠考虑. 如果将 utf8 转变为变定长编码. GBK. GB2312 可能更好.

3. 数据计算
$P_i = \frac{\text{以 } S_i \text{ 开始的序列}}{\text{以所有序列}}$
$TM_{ij} = \frac{\text{从 } S_i \to S_j \text{ 的转移}}{\text{从 } S_i \text{ 的所有转移}}$
1° 计算 P_i 时, 只用关心 statefile.txt 的每一行的第一个字符即可.
2° 计算 TMatrix
对每个转移状态 BE BM SS SE…
计数 count[0], count[1] … count[7].
(使用 map 与 switch)
转移矩阵的每一行为1
3° 计算 CMatrix
① 建立数组 二维
列: 所有中文编码的总字符
行: 4个状态

数据准备
1° 载入 inputfile.utf8
2° 文本标记
~~今/B天/E是/S星/B期/M一~~
今/B天/E是/S星/B期/M一/E
状态序列 ↓ BESBME
3° 输出 markfile.txt
输出 statefile.txt
eg. BESBME

数据计算
初始概率 Pi=[S B M E]
状态转移矩阵 TMatrix[M][M]
混淆矩阵 CMatrix[N][N]

解码分析
输入序列 "武汉的天气很热"
输出序列 "武汉 的 天气 很 热"
BESBESS

② 使用 map.
1? 对 markfile.txt 中的每一个中文字符 插入到一个 map<string, int> 中, int 以递增的形式作为该字符的一个标号(实际上为 CMatrix 的列下标)
2? 再对 markfile 中进行统计计数 "今/B". "天/E" 然后在相应位置加1.
(使用 map 插值、查找、switch 实现)

4. 解码分析

1° 如果是采用简单的二维数组

1> 将输入序列转化为编码的码字序列

2> 将码字序列作为观察序列载入模型

3> 得到解码序列

⇔ 根据 0→S；1→B；2→M；3→E

4 根据S与E进行分词

2° 如果是采用 Map

1> 查找训练时(即计算CMatrix)建立的map

得到对应的索引值

2> 将索引值作为观察值载入模型

⋮

* 使用Map存在一个很严重的问题

如：魑魅魍魉是妖怪

结果训练时，并没有魑魅魍魉，即查找不到魑魅魍魉对应的索引值

这样观察序列就要另作处理。