```
# -*- coding: utf-8 -*-
# @Project_Name : algorithm
# @File         : cut_test.py
# @Author       : Licartu
# @Time         : 2018-10-17 12:31

def load_model(f_name):
    ifp = open(f_name, 'rb').read()
    ifp = ifp.decode('utf-8', "ignore")
    return eval(ifp)


import chardet

file = open("prob_emit.py", 'rb').read()
f_charInfo = chardet.detect(file)
print(f_charInfo)
prob_start = load_model("prob_start.py")
prob_trans = load_model("prob_trans.py")
prob_emit = load_model("prob_emit.py")

# 维特比算法，一种递归算法
def viterbi(obs, states, start_p, trans_p, emit_p):
    V = [{}]
    path = {}
    print("obs:",obs)
    for y in states:    # 初始值
        V[0][y] = start_p[y] * emit_p[y].get(obs[0], 0) # 在位置 0，以 y 状态为末尾的状态序列的最大概率
        path[y] = [y]
    print("V: ",V)
    print("path: ",path)
    for t in range(1, len(obs)):
        V.append({})
        newpath = {}
        for y in states:    # 从 y0 -> y 状态的递归
            (prob, state) = max([(V[t - 1][y0] * trans_p[y0].get(y, 0) * emit_p[y].get(obs[t], 0), y0) for y0 in states if V[t - 1][y0] > 0])
            V[t][y] = prob
            newpath[y] = path[state] + [y]
        path = newpath    # 记录状态序列
    (prob, state) = max([(V[len(obs) - 1][y], y) for y in states])    # 在最后一个位置，以 y 状态为末尾的状态序列的最大概率
    print("state:",state)
```

```
        print("V: ",V)
        print("path:",path)
        return prob, path[state]    # 返回概率和状态序列


def cut(sentence):
        prob, pos_list = viterbi(sentence, ('B', 'M', 'E', 'S'), prob_start, prob_trans, prob_emit)
        start = 0
        end = 0
        cut_str = ""

        for state in pos_list:
                if state == "S" or state == "E":
                        end += 1
                else:
                        end += 1
                        continue
                cut_str += sentence[start:end] + " "
                start = end
        print(pos_list)
        return (prob, cut_str[:-1])

if __name__ == '__main__':
        test_sentence = u"适应环境是生物生存下来的重要原因。"
        prob,pos_list = cut(test_sentence)
        print("prob：    ",prob)
        print("pos_list: ",pos_list)
```