组　　号＿第一组＿＿＿　　　　　　　　　　　　　　　　　　　　密级＿公开＿

年　　份＿2024＿＿　　　　　　　　　　　　　　　　　　　　　　成绩＿＿＿＿

福州大学

系统综合实践课程实验报告

实验四 隐马尔可夫链标注词性

所 在 院 系：　　计算机与大数据学院

专 业 年 级：　　21 级计算机科学与技术(实验班)

学　　　 号：　　102101141

姓　　　 名：　　高孙炜

任 课 教 师：　　刘菀玲

联 系 方 式：　　13599396788

邮 件 地 址：　　1845613403@qq.com

1 实验介绍

隐马尔科夫链也称之为Hidden Markov Model即HMM，是一种统计模型，广泛应用在语音识别，词性标注，音字转换，概率文法等各个自然语言处理等应用领域。经过长期发展，尤其是在语音识别中的成功应用，使它成为一种通用的统计工具。

2 实验环境

1. MindSpore1.1.1+Python3.7
2. scipy1.14.1
3. numpy1.18.4

3 实验流程

3.1 数据初始化

```
import os
file_path = os.path.join(os.getcwd(),
"\own\idi_soft\idi_development\codes\python\lab1\corpus.txt")
start_c = {}
transport_c = {}
emit_c = {}
Count_dic = {}
state_list = ['Ag', 'a', 'ad', 'an', 'Bg', 'b', 'c', 'Dg', 'd', 'e', 'f', 'h',
'i', 'j', 'k', 'l', 'Mg', 'm',
              'Ng', 'n', 'nr', 'ns', 'nt', 'nx', 'nz', 'o', 'p', 'q', 'Rg', 'r',
's', 'na', 'Tg', 't', 'u', 'Vg'
              , 'v', 'vd', 'vn', 'vvn', 'w', 'Yg', 'y', 'z']
lineCount = 1
for state0 in state_list:
    transport_c[state0] = {}
    for state1 in state_list:
        transport_c[state0][state1] = 0.0
    emit_c[state0] = {}
    start_c[state0] = 0.0

vocabs = []
classify = []
```

```
class_count = {}
for state in state_list:
    class_count[state] = 0.0


with open(file_path, encoding = "utf-8") as file:
lines = file.readlines()
for line in lines:
line = line.strip()
if not line : continue
lineCount += 1
words = line.split(" ")
for word in words:
    position = word.index('/')
    if '[' in word and ']' in word:
        vocabs.append(word[1:position])
        vocabs.append(word[position + 1 : -1])
        break
    if '[' in word:
        vocabs.append(word[1:position])
        classify.append(word[position + 1 :])
        break
    if ']' in word:
        vocabs.append(word[:position])
        classify.append(word[position + 1 : -1])
        break

    vocabs.append(word[:position])
    classify.append(word[position + 1:])

if len(vocabs) != len(classify):
    print('词汇数量与类别数量不一致')
    break

else:
    for n in range(0, len(vocabs)):
        class_count[classify[n]] += 1.0
        if vocabs[n] in emit_c[classify[n]]:
            emit_c[classify[n]][vocabs[n]] += 1.0
        else:
            emit_c[classify[n]][vocabs[n]] = 1.0

        if n == 0:
            start_c[classify[n]] += 1.0

        else:
            transport_c[classify[n - 1]][classify[n]] += 1.0

vocabs = []
classify = []

for state in state_list:
    start_c[state] = start_c[state] * 1.0 / lineCount
    for li in emit_c[state]:
        emit_c[state][li] = emit_c[state][li] / class_count[state]
    for li in transport_c[state]:
        transport_c[state][li] = transport_c[state][li] / class_count[state]
```

3.2 Viterbi算法解码

```
def hmm_viterbi(obs, states, start_p, trans_p, emit_p):
    path = {}
    V = [{}]
    for state in states:
        V[0][state] = start_p[state] * emit_p[state].get(obs[0], 0)
        path[state] = [state]
    for n in range(1, len(obs)):
        V.append({})
        newpath = {}
        for k in states:
            pp, pat = max([(V[n - 1][j] * trans_p[j].get(k, 0) *
emit_p[k].get(obs[n], 0), j) for j in states])
            V[n][k] = pp
            newpath[k] = path[pat] + [k]
        path = newpath
    (prob, state) = max([(V[len(obs) - 1][y], y) for y in states])
    return prob, path[state]
```

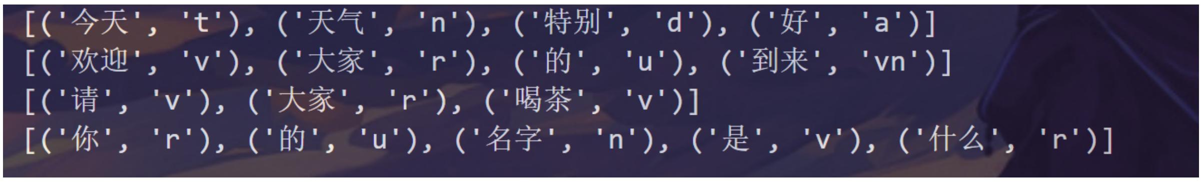
3.3 测试

```
test_strs = ["今天 天气 特别 好", "欢迎 大家 的 到来", "请 大家 喝茶", "你 的 名字 是 什
么"]

for li in range(0, len(test_strs)):
    test_strs[li] = test_strs[li].split()

for li in test_strs:
    p, out_list = hmm_viterbi(li, state_list, start_c, transport_c, emit_c)
    print(list(zip(li, out_list)))
```

实验结果如图



```
[('今天', 't'), ('天气', 'n'), ('特别', 'd'), ('好', 'a')]
[('欢迎', 'v'), ('大家', 'r'), ('的', 'u'), ('到来', 'vn')]
[('请', 'v'), ('大家', 'r'), ('喝茶', 'v')]
[('你', 'r'), ('的', 'u'), ('名字', 'n'), ('是', 'v'), ('什么', 'r')]
```

4 实验总结

本实验使用隐马尔科夫链求解词性标注，同时了解了Viterbi算法