一.运行环境

用anaconda的Jupiter lab或者Jupiter notebook直接运行即可，主要代码的首行都写有该段代码的注释，按照顺序一步步执行即可。

二.操作步骤

1.获取数据并预处理

先读取dict存入dic列表，对应以下代码

[1]:
```
#读取dict并存入列表
fdict = open("corpus.dict.txt",encoding="utf-8")
dic = []
fdictlines = fdict.readlines()
del(fdictlines[0])
for eachline in fdictlines:
    eachline = eachline.strip('\n')
    dic.append(eachline)

fdict.close()
```

再同理读取文章存入sentence列表，对应以下代码

[2]:
```
#读取文章
fsentence = open("corpus.sentence.txt",encoding="utf-8")
fsentencelines = fsentence.readlines()
sentence = []
for eachline in fsentencelines:
    eachline = eachline.strip('\n')
    sentence.append(eachline)

fsentence.close()
```

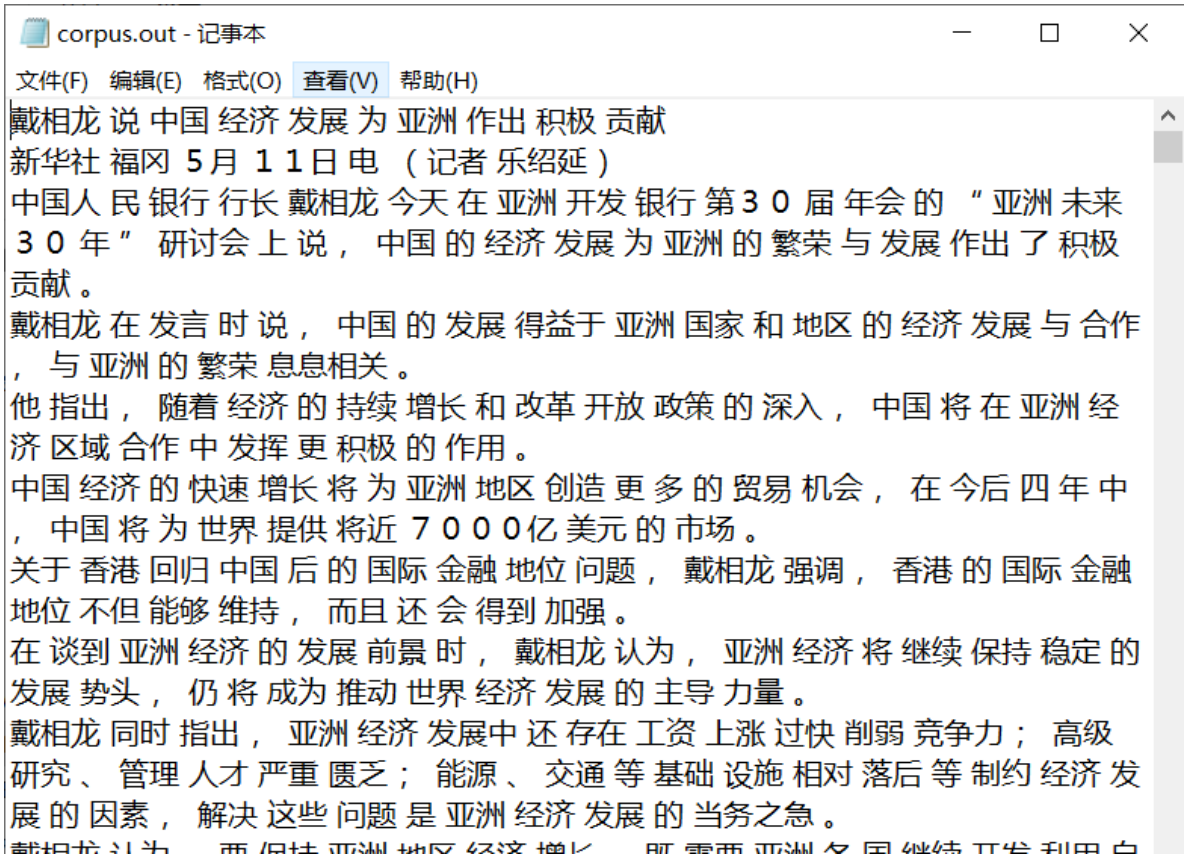
可以直接查看dic和sentence列表内容

2.最大匹配法

用最大匹配法分句，对应以下代码

[ ]:
```
#用最大匹配法分句，一行行处理并一行行写入answer.out.txt

fout=open("corpus.out.txt", mode='w+', encoding="utf-8")

maxlen = 10

num = len(sentence)
#一行行分词然后一行行写入,像一个切香肠的过程
for i in range(0,num):

    #分词存入ans列表
    onesentence = sentence[i]

    ans = []
    length = len(onesentence)
    while length>0 :
        substring = onesentence[0:maxlen]
        while substring not in dic:
            if len(substring)==1:
                 break
            substring = substring[0:len(substring)-1]
        ans.append(substring)
        onesentence = onesentence[len(substring):]
        length = len(onesentence)



    #写入文件
    for j in range(len(ans)):
        print(ans[j])
        fout.write(ans[j])
        fout.write(" ")
  #  fout.write(ans[-1])
  #  print(ans[-1])
    fout.write("\n")
    ans.clear()

fout.close()
```

然后可以得到corpus.out.txt的结果



3.分词算法评价

先读取要比较的两个文件，对应以下代码

[5]:
```
#读取两个文件存成fanswerlines和foutlines

fanswer = open("corpus.answer.txt",encoding="utf-8")
fout = open("corpus.out.txt",encoding="utf-8")


fanswerlines = fanswer.readlines()
foutlines = fout.readlines()


fanswer.close()
fout.close()
```

初始化全局三个变量，对应以下代码

[6]:
```
#接下来要对fanswerlines和foutlines逐行比较，计算准确数，answer词个数和out词个数，以此计算P，R，F
#初始化
accuracynum = 0

answernum = 0

outnum = 0
```

用双指针法计算每行相同个数，对应以下代码

[7]:
```
#用双指针法完成每一行的比较
for i in range(0,200):
    a = fanswerlines[i].split( )       # 以空格为分隔符
    b = foutlines[i].split( )

    tempa = 0 #记录目前为止a读的个数
    tempb = 0 #记录目前为止b读的个数
    count = 0 #记录正确个数

    la = 0 #a当前列表元素序号
    lb = 0 #b当前列表元素序号

    lengtha = len(a) #列表a单词个数
    lengthb = len(b) #列表b单词个数


    while(la<lengtha and lb<lengthb):
        tempa+=len(a[la])
        tempb+=len(b[lb])
        if a[la] == b[lb]:
            count+=1
        if tempa == tempb:
            la+=1
            lb+=1
            continue
        elif tempa<tempb:
            la+=1
            tempb-=len(b[lb])
            continue
        else:
            lb+=1
            tempa-=len(a[la])
            continue
    accuracynum+= count

    answernum+= lengtha

    outnum+= lengthb
```

输出PRF的答案并存储到corpus.prf.txt中，代码和结果如下

```
    print("第"+str(i+1)+"行的P,R,F: ")
    pi=float(count/lengthb)
    ri=float(count/lengtha)
    fi=2*pi*ri/(pi+ri)

    prf.write("第"+str(i+1)+"行的P,R,F: ")
    prf.write('%.3f' % pi)
    prf.write(" ")
    prf.write('%.3f' % ri )
    prf.write(" ")
    prf.write('%.3f' % fi )
    prf.write("\n")

    print("Precision = "+str(count)+"/"+str(lengthb)+" = "+str(pi))

    print("Recall= "+str(count)+"/"+str(lengtha)+" = "+str(ri))

    print("F = "+"2*p*r/(p+r) = "+str(fi))

prf.close()
```

```
第1行的P,R,F：  1.000 1.000 1.000
第2行的P,R,F：  1.000 1.000 1.000
第3行的P,R,F：  0.949 0.949 0.949
第4行的P,R,F：  1.000 1.000 1.000
第5行的P,R,F：  1.000 1.000 1.000
第6行的P,R,F：  1.000 1.000 1.000
第7行的P,R,F：  1.000 1.000 1.000
第8行的P,R,F：  1.000 1.000 1.000
第9行的P,R,F：  1.000 1.000 1.000
第10行的P,R,F：  1.000 1.000 1.000
第11行的P,R,F：  1.000 1.000 1.000
第12行的P,R,F：  1.000 1.000 1.000
第13行的P,R,F：  0.909 0.909 0.909
第14行的P,R,F：  1.000 1.000 1.000
第15行的P,R,F：  1.000 1.000 1.000
```

最后计算总的200行prf的值并输出，对应以下代码

[8]:
```
#输出前200行的正确个数以及前200行的answer单词数以及前200行的out单词数
print("前200行正确数：")
print(accuracynum)
print("前200行answer单词数：")
print(answernum)
print("前200行out单词数：")
print(outnum)
print('\n')
p = float(accuracynum/outnum)
r = float(accuracynum/answernum)
f = 2*p*r/(p+r)
print("前200行Precision = "+str(accuracynum)+"/"+str(outnum)+" = "+str(p))

print("前200行Recall= "+str(accuracynum)+"/"+str(answernum)+" = "+str(r))

print("前200行F = "+"2*p*r/(p+r) = "+str(f))
```

```
前200行正确数：
5250
前200行answer单词数：
5272
前200行out单词数：
5273


前200行Precision = 5250/5273 = 0.99563815664707
前200行Recall= 5250/5272 = 0.9958270106221547
前200行F = 2*p*r/(p+r) = 0.995732574679943
```