CS practice project report template

Project Name:spyder菜谱

Your Name:蒋伯源

# Problem definition

## 1.1 What is the project about? (What is it? Why you want to do it? Possible applications)

This project is used for getting the ingredient lists of different cooking styles like Chinese food and western food. This project can also present the frequencies of the ingredients and list them from the most frequent to less frequent ones.

I want to do this project because I love eating food from all over the world, and I want to know the differences between the food from different regions of the world.

This project can possibly be applicated to solve some problem regarding different cultures, for example, this project can be used to find the main food used in a region, and these data can then be used for deducing the region’s culture or climate or so on.

## 1.2 What are the success criteria? (What are your goals that will guide you through design and implementation and can be evaluated)

Can generate a word cloud that shows the frequencies of ingredients being used.

Can be applied to different URL under a specific Website

# Design overview

How to solve the problem in part 1? You need to design your software system, design about the input, process, and output parts.

Please list the Identifier tables if needed.

INPUT:

The Url

PROCESS

Describe your key algorithms in flowchart or pseudocode.

Key algorithm 1:

'''获取每道菜的连接'''

FUNCTION GetWeb(Page,Url):

WebList=[]

for i in range(1,Page):

URL=Url+'?&page='+str(i)

Resp=request.urlopen(URL)

html\_data=Resp.read().decode('utf-8')

Soup=bs(html\_data,'html.parser')

Web=Soup.find\_all('a',target='\_blank')

for i in Web[0:len(Web)-20]:

WebList.append(i['href'])

ENDFOR

ENDFOR

RETURN WebList

ENDFUNCTION

Key algorithm2:

'''获取每道菜的材料'''

FUNCTION GetIngre(WebList):

IngreList=[]

for i in WebList:

URL=i

try:

Resp=request.urlopen(URL)

html\_data=Resp.read().decode('utf-8')

Soup=bs(html\_data,'html.parser')

Ingre=Soup.find\_all('h4')

for i in Ingre[1:]:

IngreList.append(i.find\_all(text=True)[0])

ENDFOR

except request.HTTPError:

continue

ENDFOR

RETURN IngreList

ENDFUNCTION

Key algorithm 3:

'''清理数据'''

FUNCTION CleanIngre(Ingre):

pattern <- re.compile(r'[\u4e00-\u9fa5]+')

filterdata <- re.findall(pattern, Ingre)

CleanedIngre <- ''.join(filterdata)

RETURN CleanedIngre

ENDFUNCTION

Key algorithm 4:

'''生成词频字典'''

FUNCTION GetFreq(IngreList):

Ingre\_df=pd.DataFrame({'Ingre':IngreList})

Ingre\_stat=Ingre\_df.groupby(Ingre\_df['Ingre'])

Ingre\_stat=Ingre\_stat['Ingre'].agg([numpy.size])

Ingre\_stat=Ingre\_stat.reset\_index()

Ingre\_frequence <- {x[0]:x[1] for x in Ingre\_stat.head(1000).values}

ENDFOR

for i in DeleteList:

del(Ingre\_frequence[i])

ENDFOR

RETURN Ingre\_frequence

ENDFUNCTION

Key algorithm 5:

'''主程序生成云图'''

FUNCTION Main(Page,Url):

WebList=GetWeb(Page,Url)

IngreList=GetIngre(WebList)

Ingre=''

for i in IngreList:

Ingre+=i

ENDFOR

CleanedIngre=CleanIngre(Ingre)

IngreList=jieba.lcut(CleanedIngre)

Ingre\_frequence=GetFreq(IngreList)

wordcloud=WordCloud(font\_path="simhei.ttf",background\_color="white",max\_font\_size=80)

wordcloud=wordcloud.fit\_words(Ingre\_frequence)

plt.imshow(wordcloud)

RETURN Ingre\_frequence

ENDFUNCTION

……

OUTPUT:

A word Cloud

# Implementation

List the key techniques used in your software, like arrays, searching, sorting, dictionaries, python modules (third-party, like matplotlib, numpy, scipy, sklearn etc.)

Tech #1:Searching

Why you use tech 1 in your project?

First create a list of unwanted words and then find if these words are in the ingredient list, if so, delete them

How do you use it? Put your code screenshots here.

for i in DeleteList:

del(Ingre\_frequence[i])

Tech #2:dictionaries

Why you use tech 2 in your project?

Dictionary is convenient for managing data, and a dictionary is required to generate a word cloud

How do you use it? Put your code screenshots here.

def GetFreq(IngreList):

Ingre\_df=pd.DataFrame({'Ingre':IngreList})

Ingre\_stat=Ingre\_df.groupby(Ingre\_df['Ingre'])

Ingre\_stat=Ingre\_stat['Ingre'].agg([numpy.size])

Ingre\_stat=Ingre\_stat.reset\_index()

Ingre\_frequence = {x[0]:x[1] for x in Ingre\_stat.head(1000).values}

for i in DeleteList:

del(Ingre\_frequence[i])

return Ingre\_frequence

Tech #3:array

Why you use tech 3 in your project?

It is convenient to use an array to store data.

How do you use it? Put your code screenshots here.

IngreList=[]

for i in WebList:

URL=i

try:

Resp=request.urlopen(URL)

html\_data=Resp.read().decode('utf-8')

Soup=bs(html\_data,'html.parser')

Ingre=Soup.find\_all('h4')

for i in Ingre[1:]:

IngreList.append(i.find\_all(text=True)[0])

except request.HTTPError:

continue

return IngreList

Tech #4:urllib&bs4

Why you use tech 4 in your project?

They are useful when I want ton get data from a website

How do you use it? Put your code screenshots here.

def GetWeb(Page,Url):

WebList=[]

for i in range(1,Page):

URL=Url+'?&page='+str(i)

Resp=request.urlopen(URL)

html\_data=Resp.read().decode('utf-8')

Soup=bs(html\_data,'html.parser')

Web=Soup.find\_all('a',target='\_blank')

for i in Web[0:len(Web)-20]:

WebList.append(i['href'])

return WebList

'''获取每道菜的材料'''

def GetIngre(WebList):

IngreList=[]

for i in WebList:

URL=i

try:

Resp=request.urlopen(URL)

html\_data=Resp.read().decode('utf-8')

Soup=bs(html\_data,'html.parser')

Ingre=Soup.find\_all('h4')

for i in Ingre[1:]:

IngreList.append(i.find\_all(text=True)[0])

except request.HTTPError:

continue

return IngreList

Tech #4:jieba&pandas&numpy

Why you use tech 4 in your project?

These three module can be used together to generate the dictionary I need. Jieba can divide words in Chinese, and pandas and numpy can help to count the frequencies.

How do you use it? Put your code screenshots here.

'''生成词频字典'''

def GetFreq(IngreList):

Ingre\_df=pd.DataFrame({'Ingre':IngreList})

Ingre\_stat=Ingre\_df.groupby(Ingre\_df['Ingre'])

Ingre\_stat=Ingre\_stat['Ingre'].agg([numpy.size])

Ingre\_stat=Ingre\_stat.reset\_index()

Ingre\_frequence = {x[0]:x[1] for x in Ingre\_stat.head(1000).values}

for i in DeleteList:

del(Ingre\_frequence[i])

return Ingre\_frequence

'''主程序生成云图'''

def Main(Page,Url):

WebList=GetWeb(Page,Url)

IngreList=GetIngre(WebList)

Ingre=''

for i in IngreList:

Ingre+=i

CleanedIngre=CleanIngre(Ingre)

IngreList=jieba.lcut(CleanedIngre)

Ingre\_frequence=GetFreq(IngreList)

wordcloud=WordCloud(font\_path="simhei.ttf",background\_color="white",max\_font\_size=80)

wordcloud=wordcloud.fit\_words(Ingre\_frequence)

plt.imshow(wordcloud)

return Ingre\_frequence

Tech #5:wordcloud&matplotlib

Why you use tech 4 in your project?

The can help me visualize my data

How do you use it? Put your code screenshots here.

wordcloud=WordCloud(font\_path="simhei.ttf",background\_color="white",max\_font\_size=80)

wordcloud=wordcloud.fit\_words(Ingre\_frequence)

plt.imshow(wordcloud)

# Results

Show the outcomes/results of your projects

**粤菜**

****

****

**川菜**

# Evaluation and conclusion

Evaluate the project based on your success criteria in Part 1.

The project meets all the criteria.

What is the conclusion of the project?

**Different regions all have their own flavor of food and different ingredients to make it.**

# References

List all the materials you referred to.

<https://segmentfault.com/a/1190000010473819>

<https://www.cnblogs.com/A9kl/p/9311246.html>

<https://ask.hellobi.com/blog/yuguiyang1990/9445>

<https://www.jianshu.com/p/42f1d2909bb6>

<https://blog.csdn.net/qq_18649781/article/details/89033749>

<https://www.baidu.com/link?url=0Ef1Xy43_nVbVZAu01trk3CBMKtkb5MPEPGYhx7ejyMmkfeuodhALhZu_4Kokn5CZCHkKxMyj6ch3tXKFRCrHTeUefh7hijs7153aCsJ7CS&wd=&eqid=9498d611000b6263000000025d94b16b>

<https://blog.csdn.net/Chenftli/article/details/82887024>

<https://blog.csdn.net/u012706792/article/details/80892510>

<https://blog.csdn.net/fanfanyuzhui/article/details/78503608>

<https://blog.csdn.net/u013317445/article/details/85268877>

<https://blog.csdn.net/qq_39516859/article/details/81607889>

<https://blog.csdn.net/qq_25436597/article/details/79079435>

<https://segmentfault.com/a/1190000012394176>

<https://www.runoob.com/regexp/regexp-syntax.html>

<https://blog.csdn.net/wxystyle/article/details/76121270>