**1、引言**

编写目的

输入任意关键词，然后在ｐｕｂｍｅｄ上查询相应得文献ＩＤ，爬取每个文献的＇Pubdate','Source','Author','Title','PubType','FullJournalName','Abstract','ISSN','ImpactFactor'，

然后把这些信息生成报表

**2、总体设计**

2.1需求规定

　　输入任意位于关键词和结果的输出目录，

2.2运行环境

　本软件运行于Windows7及以上Windows版本

2.3基本设计概念和处理流程

　利用ｐｕｂｍｅｄ的ＡＰＩ接口进行查询搜索获取相关文献的信息，爬取到文献对应的ＩＳＳＮ后再依据已有的２０１３年ＳＣＩ杂志影响因子列表对文献进行注释

2.4设计约束

　 由于本地的２０１３年ＳＣＩ影响因子列表内容太多，所以导致循环判断所花费的时间比较多

2.5人工处理过程

需要人工输入搜索的文献关键词和结果输出目录

**3、接口设计**

3.1用户接口

3.2外部接口

3.3内部接口

**4、运行设计**

4．1运行模块组合

4.2运行控制

4.3运行时间

**5、系统论据结构设计**

5.1逻辑结构设计要点

5.2物理结构设计要点

5.3数据结构与程序的关系

**6、系统出错处理设计**

6.1出错信息

6.2补救措施

6.3系统维护设计

# -\*- coding: utf-8 -\*-

"""

自动获取NCBI文献数据库（PubMed）的文献的软件，指定文献参数，

用Python访问API获取文献ID列表，再通过访问ID对应的文献网页，

用爬虫获取文献信息，整理出报表，报表内容包括（'PMID','Pubdate','Source','Author',

'Title','PubType','FullJournalName','Abstract','ISSN','ImpactFactor'）

@author: khe

"""

from urllib import request

from bs4 import BeautifulSoup

import re

import time

import xlsxwriter as xlsx

import xlrd

import urllib

def sort(keywords):#对输入的相关参数进行整理，避免空格，若出现空格则以+号取代

para = keywords.split()

keywords\_sorted = ''

for i in range(len(para)):

if i == len(para) - 1:

keywords\_sorted = keywords\_sorted + para[i]

break

keywords\_sorted = keywords\_sorted + para[i] + '+'

return keywords\_sorted

class Tool:

#利用正则表达式，匹配文献网页的相关元素

def \_\_init\_\_(self):

self.Id\_re = re.compile('<Id>(.\*?)</Id>')

self.Pubdate\_re = re.compile('<Item Name="PubDate" Type="Date">(.\*?)</Item>')

self.Source\_re = re.compile('<Item Name="Source" Type="String">(.\*?)</Item>')

self.Author\_re = re.compile('<Item Name="Author" Type="String">(.\*?)</Item>')

self.Title\_re = re.compile('<Item Name="Title" Type="String">(.\*?)</Item>')

self.PubType\_re = re.compile('<Item Name="PubType" Type="String">(.\*?)</Item>')

self.FullJournalName\_re = re.compile('<Item Name="FullJournalName" Type="String">(.\*?)</Item>')

def replace(self,content):

Id = re.findall(self.Id\_re,content)

Pubdate = re.findall(self.Pubdate\_re,content)

Source = re.findall(self.Source\_re,content)

Author = re.findall(self.Author\_re,content)

Title = re.findall(self.Title\_re,content)

PubType = re.findall(self.PubType\_re,content)

FullJournalName = re.findall(self.FullJournalName\_re,content)

url = 'https://www.ncbi.nlm.nih.gov/pubmed/?term=' + str(Id[0])

#新建url连接去爬取文献的摘要信息

try:

request = urllib.request.Request(url)

reponse = urllib.request.urlopen(request)

content = reponse.read().decode('utf-8')

except urllib.error.URLError as Error:

print(Error)

soup = BeautifulSoup(content,'lxml')

t = soup.find\_all('abstracttext')

pattern = re.compile('<abstract.\*?>|</abstracttext>')

Abstract = re.sub(pattern,'\n',str(t))

url1='https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi?db=pubmed&id=' + str(Id[0])

#新建url连接去爬去文献对应得ISSN编号信息

try:

request1 = urllib.request.Request(url1)

reponse1 = urllib.request.urlopen(request1)

content1 = reponse1.read().decode('utf\_8')

except urllib.error.URLError as Error:

print(Error)

issn\_re = re.compile('<Item Name="ISSN" Type="String">(.\*?)</Item>')

ISSN = str(re.findall(issn\_re,content1))

#根据爬取的文献的ISSN信息注释其相应的影响因子信息

data = xlrd.open\_workbook('2013年SCI杂志影响因子列表.xls')

#读入本地的一个影响因子列表,这个文件就放在python的当前工作目录下

table = data.sheet\_by\_index(0)

nrows = table.nrows

#如果没有抓到ISSN，则影响因子为空

print(ISSN)

ImpactFactor = ''

for i in range(nrows):

cell = '[\'' + table.col(2)[i].value + '\']'

if cell == ISSN:

ImpactFactor = table.col(4)[i].value

break

print(cell)

print(ImpactFactor)

list\_value = [Id,Pubdate,Source,Author,Title,PubType,FullJournalName,Abstract,ISSN,ImpactFactor]

return list\_value

class Literature:

#通过相关参数在pubmed中获取相应的文献信息

def \_\_init\_\_(self,url,search,summary,info\_sorted,tool):

#info\_sorted是经过整理得到的相关参数

#tool网页的正则匹配工具

#listing建立excel表格，将获取文献信息导入表格

self.url = url

self.search = search

self.summary = summary

self.info\_sorted = info\_sorted

self.tool = tool

self.Esearch = url + search + str(info\_sorted)

def get\_uid(self):

#访问pubmed的API，匹配其中的UID,返回匹配的UID个数

try:

request = urllib.request.Request(self.Esearch)

reponse = urllib.request.urlopen(request)

content = reponse.read().decode('utf-8')

except urllib.error.URLError as Error:

print(Error)

patterns = re.compile('<Id>(.\*?)</Id>')

self.Id\_s = Id\_s = re.findall(patterns,content)

print('文献ID已成功匹配\n')

length = len(Id\_s)

return length

def info\_ready(self,i):

Esummary\_b = self.url + self.summary

Esummary = Esummary\_b + str(self.Id\_s[int(i)])

try:

request = urllib.request.Request(Esummary)

reponse = urllib.request.urlopen(request)

content = reponse.read().decode('utf-8')

except urllib.error.URLError as Error:

print(Error)

return content

def get\_info(self,element):

value = self.tool.replace(element)

return value

class List:

#创建Excel表格，并导入相关的文献信息

def create(self,path,Litera,num):

self.path = path

self.wb = xlsx.Workbook(str(self.path))

self.ws = self.wb.add\_worksheet('Literature\_info')

list\_create = ['PMID','Pubdate','Source','Author',\

'Title','PubType','FullJournalName','Abstract','ISSN','ImpactFactor']

for i in range(len(list\_create)):

self.ws.write(0,i,list\_create[i])

for i in range(num):

element = Litera.info\_ready(i)#网页全部信息

values = Litera.get\_info(element)#通过匹配得到的所需信息

for k in range(len(values)):#信息顺序录入表格

self.ws.write(i+1,k,str(values[k]))

print('已成功录入%d条文献信息'%(i+1))

time.sleep(0.5)

self.wb.close()

def main():

keywords = input('请输入文献搜索的关键词:\n')

url = 'https://eutils.ncbi.nlm.nih.gov/entrez/eutils/'

search = 'esearch.fcgi?db=pubmed&term='

summary = 'esummary.fcgi?db=pubmed&id='

path = input('please input the file path:\n')

listing = List()

tool = Tool()

Sorted = sort(keywords)

Litera = Literature(url,search,summary,Sorted,tool)

Id\_num = Litera.get\_uid()

listing.create(path,Litera,Id\_num)

print('录入完毕\n')

if \_\_name\_\_ == "\_\_main\_\_":

main()