# 1、系统概述

**1.1、系统简介**

**提供可视化特定的社群信息（以股市资讯为例）的词频和日期关系，提供给用户发掘宏观事件之间驱动关系的辅助工具。通过统计分词并统计词频，并以时间为词频的关联特征生成词频与日期的二维统计图。另外生成制定关键词之间的简单卷积结果图表。以以上二者作为可视化信息辅助用户完成事件驱动的决策。**

**强调只是辅助决策，并非确定关联性。**

**系统主要为了提供某个词的每天出现的次数与日期之间变化关系。**

**1.2、术语表**

定义系统或产品中涉及的重要术语，为读者在阅读文档时提供必要的参考信息。

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | Annc | Announcement 公告。本系统需要获取股市公告。 |
| 2 | Dict | 自定义的词频列表[]或字典{}，词语一般不会单独出现。注释中会提示该字典的存储内容。 |
| 3 | Crawl | 爬虫 |

**1.3、系统运行环境**

Win10

MySQL数据库部署

**1.4、开发环境**

**Python 2.7.15  
MySQL 8.0带python connector  
Matlab 2018a带extern**

**其中Python的额外库文件**

unicodecsv,scrapy,docx,pdfminer,setuptools,MySQL-python,pymysql

# 2、数据结构说明

本章说明本程序系统中使用的全局数据常量、变量和数据结构。

2.1、常量

用户个性化数据初始化

*#路径配置*

FILEPATH = conf.get('file', 'filepath')

LOGPATH = conf.get('file', 'logpath')

*#数据库*

DB\_HOST\_ADDRESS = conf.get('database', 'host\_address')

DB\_USER = conf.get('database', 'user')

DB\_PASSWORD = conf.get('database', 'password')

DB\_PORT = int(conf.get('database', 'port'))

*##周期长度枚举变量*

class CYCLELENTH:

TINY=7*#周*

SMALL=15*#两周 默认*

MEDIUM=30*#一个月*

LARGE=90*#一个季度*

HUGE=365*#一年*

*##分析类型枚举变量*

class ANALYTYPE:

DATE=1*#日期 默认*

DICT=2*#关键词*

*##排序选择枚举变量*

class SORTTYPE:

HOT=1*#最热 默认*

DENSITY=2*#最快*

包括数据文件名称及其所在目录，功能说明，具体常量说明等。

2.2、变量

ProcessInfo=Tk.StringVar()*#进度信息*

用于调试

YYYYMMDD类型的日期

*#已经拥有的资讯日期*

DateACK=[]*##YYYYMMDD类型*

*###词频日期信息提取函数需要的参数*

*#要爬取的日期*

DateCrawl=[]

#有爬取的

DateCrawled=[]

以下依次是某天的某公告里的关键词 词典

某天的关键词及其出现频次

DateAnncDict={}*#每天对应一堆字符串 {日期：{标题：[词，词，词]，标题：[词，词，词]}}*

DateDict *###{日期：{词:词数,词:词数}}*

用户制图要求：

*##当前用户的制图要求，默认周期30析，类型默认时间，排序选择默认最热*

UserRuir=[CYCLELENTH.SMALL,ANALYTYPE.DATE,SORTTYPE.HOT]

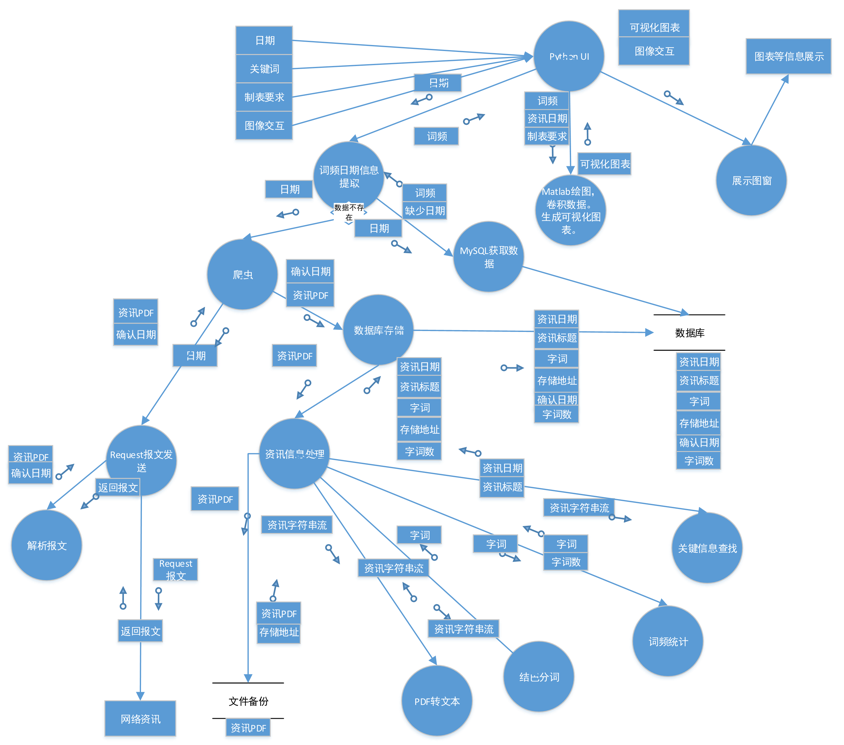
*##上次用户的制图要求*

LastUserRuir=UserRuir

列表里是三个枚举变量，见常量表

# 3、模块设计

**3.1、软件结构**

**3.2、功能设计说明**

**PythonUI是用GUI语言写的一个前台程序，可以捕捉用户操作，获取用户输入信息同时把获得的图表输出信息展示出来，并能被用户进行一定程度的操作。也作为主控函数进行如下步骤。**

**①根据用户输入和行为提供不同模式的展示信息。**

**②展示的信息来源于主控中预设的界面UI和图表信息。**

**③图表信息会从matlab对收集的数据进行处理，运算。**

**④.1收集的信息以词频信息为主，信息从数据库中获取。但是大多情况下，程序需要补充词频信息。**

**④.2通过调用爬虫函数获取新的资讯讯息。分为爬取，筛选，存储三步骤。**

**I）根据用户需求，发送报文并解析，实现资讯爬取，爬取完毕后会返回确认日期，告诉数据库当天资讯已经完整。**

**II）在筛选中预先存储资讯文件，顺便把资讯（pdf）转化为字符串，之后将字符串进行分词处理，之后对词语进行统计过滤。**

**III）最后将获取的词频信息打包存在数据库中。关注的信息只有日期和词语及其对应的统计次数。流图中大量的数据信息只是为后期提升提供支撑。**

**自此收集的信息的获取便完成。**

**另外，步骤④占用资源少，但是因为联网费时，所以独立分出线程。**

**程序中与Matlab和MySQL数据库对接时，采用线程通讯办法，即程序先打开二者的线程，然后通过二者嵌入式开发的指令与这两个线程通讯并完成操作以实现功能。**

**在进入模块设计前，该表可能需要用到。**

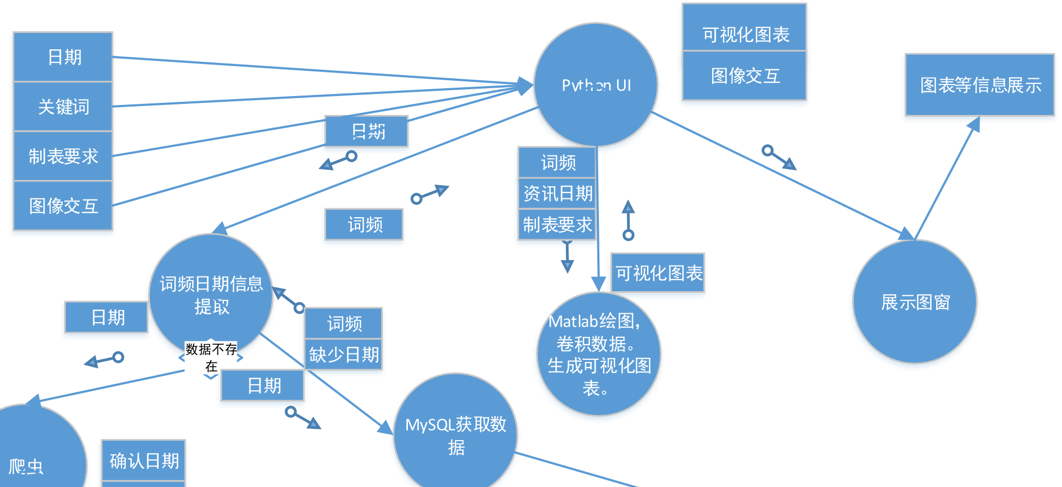
|  |  |  |
| --- | --- | --- |
| 名称 | 含义 | 数据类型 |
| 日期① | 指定日期区间段，进行数据分析。 | string |
| 关键词② | 指定两个关键词，进行数据分析。 | string |
| 制表要求③ | 寄存图表的分析周期的当前状态。 | short |
| 图像交互④ | 用户的控制信息，包括移动窗口。 | 触发信号 |
| 资讯PDF⑤ | 大多资讯的原材料，以文件路径为标识。 | .PDF |
| Request报文⑥ | 请求数据的报文 | string |
| 返回报文⑦ | 请求数据后返回的报文 | string |
| 存储地址⑧ | 资讯最终存储地址 | string |
| 资讯字符串流⑨ | 资讯PDF转化后的字符串，作为分析的关键 | string |
| 词频⑩ | 收集的数据，作为数据分析的关键元素。  作为数组存在，其成员为日期，字词，字词数。 | [string，  string,  int] |
| 资讯日期⑾ | 当前统计字词来自的资讯的日期 | string |
| 字词数⑿ | 字词统计的数量 | int |
| 字词⒀ | 当前统计的字词 | string |
| 可视化图表⒁ | 数据处理后，并以规定制成的图表 | 图表 |
| 图表等信息展示⒂ | 最后展示给用户的画面 | NA |
| 缺少日期⒃ | 数据库的数据如果不全，会返回缺少数据的日期 | string |
| 确认日期⒄ | 当某日的资讯都抓取完毕，返回确认 | string |
| 资讯标题⒅ | 资讯标题 | string |
| 字词⒆ | 某个字词 | string |
| 字词数⒇ | 某个字词对应的字词数 | int |

## 3.1、主函数



主控面板，交互窗口

3.3.1、设计图



3.3.2、功能描述

简要描述模块1的业务功能。

3.3.3、输入数据

由UI控件获得

用户键入或滑动滑块等获得日期①关键词②制表要求③

触发按钮：图像交互④

词频日期信息提取模块在收到日期后会返回词频信息

从 Matlab绘图获取 图表 得以放置在UI界面上

3.3.4、输出数据

向 词频日期信息提取模块 输入日期

向 Matlab绘图提供 词频 资讯日期，制表要求

展示图窗是一个函数只是负责布局，目前程序并未考虑将其封装

3.3.6、算法和流程

* 读取用户信息
* 初始化

*#路径配置*

FILEPATH = conf.get('file', 'filepath')

LOGPATH = conf.get('file', 'logpath')

*#数据库*

DB\_HOST\_ADDRESS = conf.get('database', 'host\_address')

DB\_USER = conf.get('database', 'user')

DB\_PASSWORD = conf.get('database', 'password')

DB\_PORT = int(conf.get('database', 'port'))

*###数据库的情况：我有那几天的数据*

*#已经拥有的资讯日期*

DateACK=[]*##YYYYMMDD类型*

*##初始化已经拥有的资讯日期*

DateACK\_Init.ACK\_Init(DateACK,DB\_HOST\_ADDRESS,DB\_USER,DB\_PASSWORD,DB\_PORT)

*#要爬取的日期*

DateCrawl=[]

*##初始化默认的爬取日期*

DateCrawl\_Init.DateCrawlInit(DateCrawl,DateACK)

因为是主函数还有大量常量定义，全局变量初始化

* 设置并布局Tkink交互窗口
* 进入交互循环
* 触发时机：
  + 用户完成日期或关键词输入后，按下词频分析按钮或关键词按钮
  + 用户调节周期，触发周期设置按钮

3.3.7、函数说明

def ToDown():

触发时机按下词频分析按钮或关键词按钮

对DateCrawl处理后

将

DateCrawl#要爬取的日期

DateAnncDict={}#每天对应一堆字符串 {日期：{标题：[词，词，词]，标题：[词，词，词]}}

DateDict ###{日期：{词:词数,词:词数}}

输入 词频日期信息提取 模块

完成后

将

DateAnncDict={}#每天对应一堆字符串 {日期：{标题：[词，词，词]，标题：[词，词，词]}}

DateDict ###{日期：{词:词数,词:词数}}

再次输入 matlab绘图 模块

返回 图表

将图表放置

完成指令

def ToShow():

触发时机触发周期设置按钮

将以下内容

##当前用户的制图要求，默认周期30析，类型默认时间，排序选择默认最热

UserRuir=[CYCLELENTH.SMALL,ANALYTYPE.DATE,SORTTYPE.HOT]

DateAnncDict={}#每天对应一堆字符串 {日期：{标题：[词，词，词]，标题：[词，词，词]}}

DateDict ###{日期：{词:词数,词:词数}}

路径配置

数据库配置

输入 matlab绘图 模块

保护一下

##上次用户的制图要求

LastUserRuir=UserRuir

返回 图表

将图表放置

完成指令

3.3.8 全局数据结构与该模块的关系

作为主函数掌控所有以下数据

常量

用户个性化数据

*#路径配置*

FILEPATH = conf.get('file', 'filepath')

LOGPATH = conf.get('file', 'logpath')

*#数据库*

DB\_HOST\_ADDRESS = conf.get('database', 'host\_address')

DB\_USER = conf.get('database', 'user')

DB\_PASSWORD = conf.get('database', 'password')

DB\_PORT = int(conf.get('database', 'port'))

枚举常量

*##周期长度枚举变量*

class CYCLELENTH:

TINY=7*#周*

SMALL=15*#两周 默认*

MEDIUM=30*#一个月*

LARGE=90*#一个季度*

HUGE=365*#一年*

*##分析类型枚举变量*

class ANALYTYPE:

DATE=1*#日期 默认*

DICT=2*#关键词*

*##排序选择枚举变量*

class SORTTYPE:

HOT=1*#最热 默认*

DENSITY=2*#最快*

全局变量

ProcessInfo=Tk.StringVar()*#进度信息*

用于调试

YYYYMMDD类型的日期

*#已经拥有的资讯日期*

DateACK=[]*##YYYYMMDD类型*

*###词频日期信息提取函数需要的参数*

*#要爬取的日期*

DateCrawl=[]

#有爬取的

DateCrawled=[]

以下依次是某天的某公告里的关键词 词典

某天的关键词及其出现频次

DateAnncDict={}*#每天对应一堆字符串 {日期：{标题：[词，词，词]，标题：[词，词，词]}}*

DateDict *###{日期：{词:词数,词:词数}}*

用户制图要求：

*##当前用户的制图要求，默认周期30析，类型默认时间，排序选择默认最热*

UserRuir=[CYCLELENTH.SMALL,ANALYTYPE.DATE,SORTTYPE.HOT]

*##上次用户的制图要求*

LastUserRuir=UserRuir

列表里是三个枚举变量，见常量表

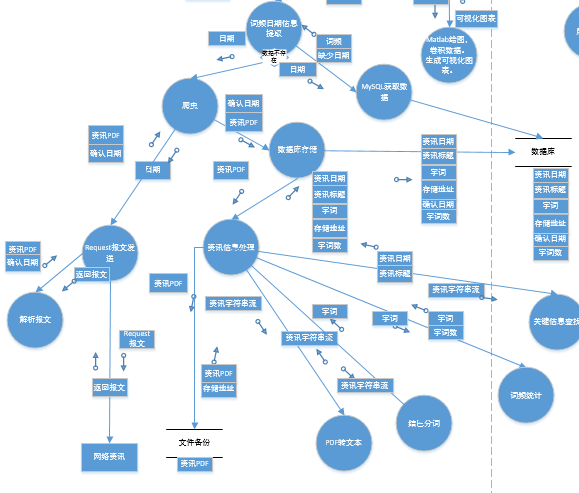
## 3.1.1、词频日期信息提取



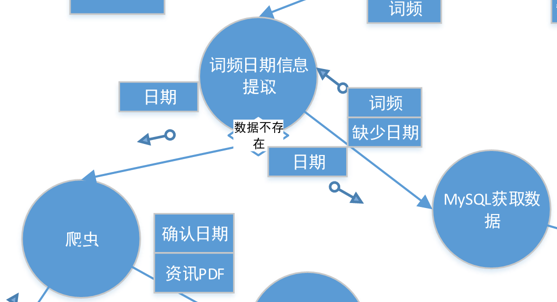
信息获取模块

又称词频日期信息提取

1、设计图



该模块很大可细分为两大部分和作为主控模块的自己



2、功能描述

获取词频信息，\*所有获取信息的主控模块\*

3、输入数据

PythonUI接入

日期①Date 和 DateCrawl

全局变量词频字典

存储路径

数据库登录信息

MySQL获取数据返回:

词频⑩

缺少日期⒃

爬虫获取数据返回:

确认日期⒄

4、输出数据

日期①输入爬虫

日期①输入

MySQL获取数据

词频⑩返回PythonUI

6、算法和流程

主函数通过DateACK和DateCrawl（已经有的数据的日期，和要获取数据的日期）取交集告诉该模块要爬取的数据DateCrawl

获得全局的词频词典

把DateCrawl输入爬虫

爬虫完成返回后

对MySQL获取模块输入Date和全局的词频词典

当模块完成操作后，就返回函数，此时全局的词频词典可能是系统要求

如果有返回缺失日期

重复以上循环

7、函数说明

def InfoSelect(*DateCrawl*,*DateAnncDict*,*DateDict*,*path*,*Date*,*MySQLinfo*):

输入DateCrawl

完成全局的词频词典操作

8 全局数据结构与该模块的关系

DateCrawl标识需要爬取的日期

DateAnncDict更新输入日期区间段的词频

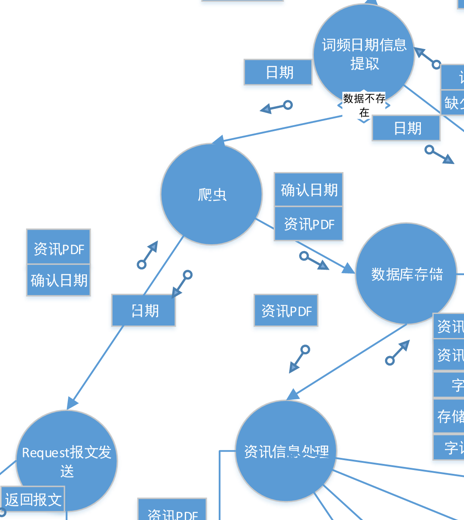
DateDict更新输入日期区间段的词频

全局词频字典：传递地址

## 3.1.1.1、爬虫



1. 设计图



2、功能描述

负责爬取PDF公告并处理公告在存到数据库里

3、输入数据

* DateCrwal要爬取的时间
* path要存储的位置
* MySQLinfo数据库登录信息

子模块都没有返回，只有结束，图示有误

4、输出数据

向数据库存储输出

* 爬取到的日期DateCrwaled
* 全局词频字典
* 数据库登录信息
* 存储地址

6、算法和流程

* 开始爬取

执行子模块

如果子模块正常返回

* 说明爬取到数据了，进行数据库存储

存储完毕，返回程序

7、函数说明

def Crwal(*DateCrwal*,*path*,*MySQLinfo*):

爬取

存储

返回

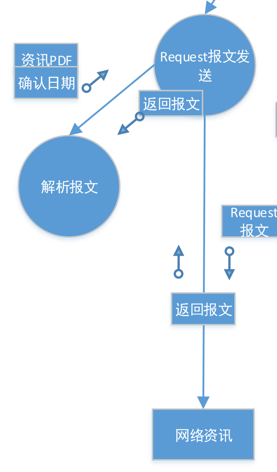
8 全局数据结构与该模块的关系

要爬取的时间：标识要爬取的日期，爬完后会去除元素

## 3.1.1.1.1 报文发送



1. 设计图



2、功能描述

* 负责发文获取pdf并存储

3、输入数据

* DateCrwal要爬取的时间
* path要存储的位置
* 解析报文返回要发起的第二批url

4、输出数据

存储文件

6、算法和流程

发送第一份报文，该报文为发现要爬取pdf的url，通过解析返回的json

可以知道该日期段的pdf url

发起第二批报文

获取所有pdf并存储

返回

7、函数说明

def RequestMaker(*DateCrawl*,*path*):

发出第一份查询公告地址的报文

解析报文

发起第二批请求下载公告的报文

返回

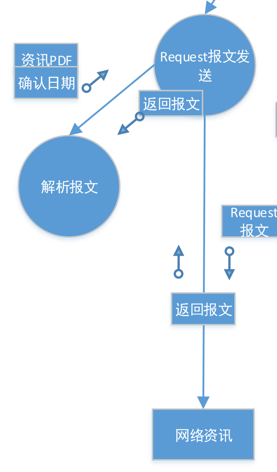
8 全局数据结构与该模块的关系

DateCrawl：标识需要爬取的日期，在第一份报文中会用到

## 3.1.1.1.1.1 解析报文



1. 设计图



2、功能描述

解析并识别报文中需要的公告的url

3、输入数据

* 报文

4、输出数据

* 要求的url

6、算法和流程

解码

正则表达式匹配

返回url

7、函数说明

def DatagramPrase(*Datagram*):

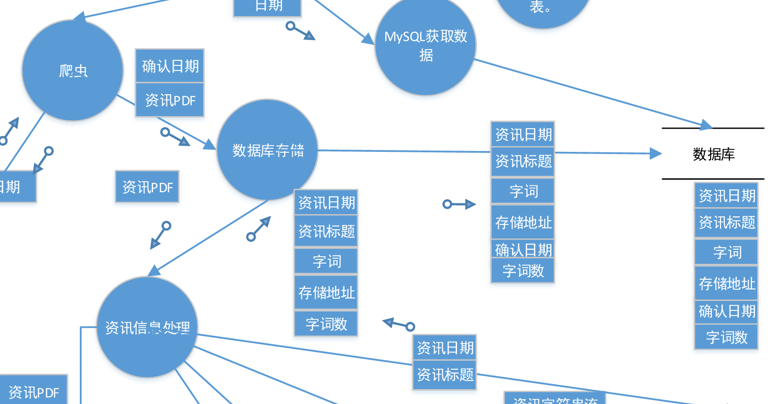
8 全局数据结构与该模块的关系

无

## 3.1.1.2 数据库存储



1. 设计图



2、功能描述

数据存入数据库

3、输入数据

* 数据库登录信息
* 爬取到日期
* 路径

资讯信息处理模块

* 返回更新好的字典

1. 输出数据

无

6、算法和流程

调用资讯信息处理模块

获得更新字典

存入数据到数据库

返回

7、函数说明

def InfoToMySQL(*SQLinfo*,*DateCrawled*=['20190531', '20190601'], #要处理的日期

*DIR*=u"F:\\2019sp\\Py\\源码\\手动爬取" , #路径):

8 全局数据结构与该模块的关系

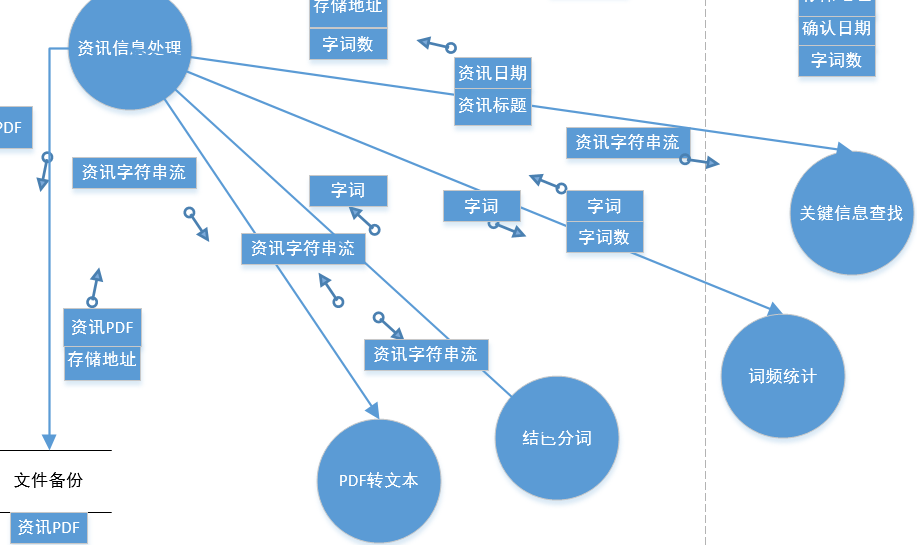
路径：读取pdf的地方

数据库登录信息：操作数据库的钥匙

## 3.1.1.2 资讯信息处理



1. 设计图



2、功能描述

按日期批量处理pdf进而查出查出资讯时间，标题,词频信息

3、输入数据

* 爬取到日期
* 路径

结巴分词

* 返回可以用的关键词

词频统计

* 返回各关键词的词频数

1. 输出数据

* 返回更新好的类似全局词频字典，供数据库写入

6、算法和流程

* 把路径进行处理，让每个公告路径映射到某个路径
* 按照日期把文件转化txt调用子模块pdf转文本
* 防止bug，读出文本调用结巴分词
* 取出词频信息，封装到每日每公告字典里
* 进行词频统计
* 获得每日字典
* 返回词典

1. 函数说明

*##读文本txt*

def text\_read(*filename*):

返回文本内字符串

*##按日期格式文件名，读取每天PDF，并存在相应文件夹*

def InfoPreprocessor(*fileDir*,*DateCrawled*):

8 全局数据结构与该模块的关系

路径：读取pdf的地方

## 3.1.1.2.1 pdf转文本



*#将一个pdf转换成txt*

def pdfTotxt(*filepath*,*outpath*):

输入读取地址，输出地址

如

F:\2019sp\Py\源码\手动爬取\20190531

F:\2019sp\Py\源码\手动爬取\20190531txt

pdf 转txt

没有返回

## 3.1.1.2.2 结巴分词



输入字符串

每75个词提取一个关键词，参考结巴算法

返回一个词列表

def WordSplit(*AnncString*):

## 3.1.1.2.3 词频统计

输入一个指定格式的字典

返回每天的词频统计



*###由{日期：{标题：[词，词，词]，标题：[词，词，词]}}*

*###获得*

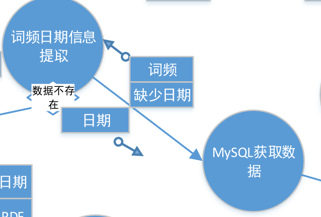
*###{日期：{词:词数,词:词数}}*

def WordCount(*DateAnncDict*):

## 3.1.1.2、MySQL获取数据



1. 设计图



2、功能描述

提取数据库内的词频信息\*提取词频信息唯一函数\*

3、输入数据

* Date读取的时间
* MySQLinfo数据库登录信息
* 全局的词频字典

1. 输出数据

向数据库提交SQL指令

返回

* 缺少的日期

6、算法和流程

根据日期访问数据库

获取信息

更新词典时带着检测日期

返回缺少的时间

7、函数说明

def MySQLGet(*Date*,*MySQLinfo*,*DateCrawl*,*DateAnncDict*):

访问数据库

更新字典

返回缺少的时间

8 全局数据结构与该模块的关系

全局词频词典：更新词典

## 3.1.2 Matlab绘图



输入数据

初始化matlab引擎

将数据丢入已经做好的算法里

获取卷积后的数

根据获取的数制图

返回图

PS代码还没完全写完需要后续补上。

# 4、 接口设计

**4.1、 用户接口**

UI操作

按键触发，输入窗口输入日期，关键词。

**5.1、外部接口**

**MySQL数据库服务**

**Matlab线程**

键盘输入，鼠标控制，显示屏显示。

网络连接及其协议。

**5.2、内部接口**

**特别说明：对输入数据库的接口在“数据库存储”模块**

**数据库输出接口只在“MySQL获取数据”模块**

**网络接口只在“Request发送报文”模块中**

其余见数据流图和模块设计

# 5、数据库设计

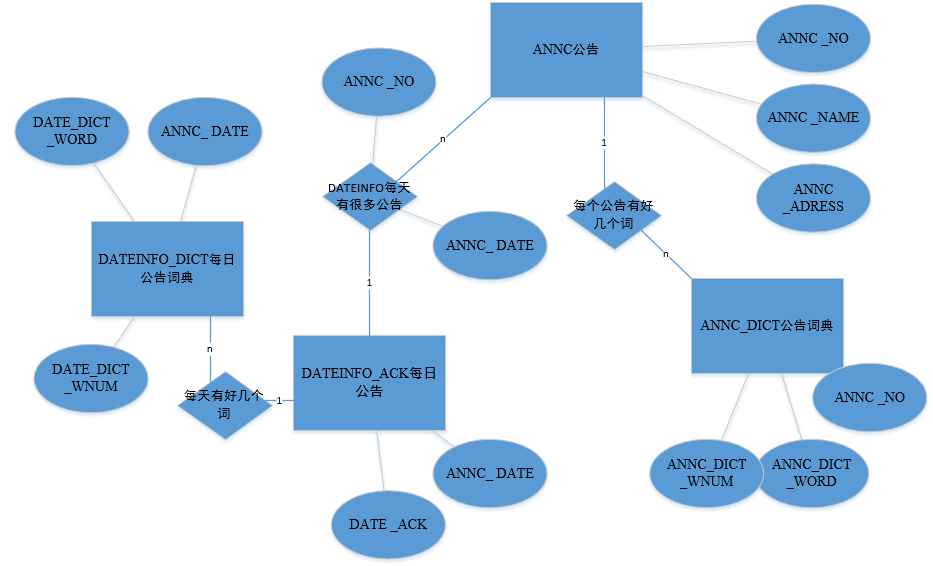
数据库名：STOCK

数据表：

|  |  |  |  |
| --- | --- | --- | --- |
| 表一：DATEINFO（含义Date每天的信息） | | | |
| 名称 | 含义 | 数据类型 | 存储长度 |
| ANNC\_ DATE | 该公告的日期 | DATE | 3 |
| ANNC \_NO | 该公告标号 | INT UNSIGNED | 4 |
| 表二：DATEINFO\_DICT（含义Date Dict 每天的词典信息） | | | |
| ANNC\_ DATE | 该公告的日期 | DATE | 3 |
| DATE\_DICT \_WORD | 该日期的某个词 | VARCHAR | 32 |
| DATE\_DICT \_WNUM | 该日期的某个词的数量 | INT UNSIGNED | 4 |
| 表三:DATEINFO\_ACK（含义：确认每天的信息是否收全） | | | |
| ANNC\_ DATE | 该公告的日期 | DATE | 3 |
| DATE \_ACK | 确认日期，意味着今日的公告爬取完了 | SMALLINT | 1 |
| 表四：ANNC（含义Announcement每个公告信息） | | | |
| 名称 | 含义 | 数据类型 | 存储长度 |
| ANNC \_NO | 该公告标号 | INT UNSIGNED | 4 |
| ANNC \_NAME | 该公告名字 | VARCHAR | 255 |
| ANNC \_ADRESS | 该公告的地址 | VARCHAR | 255 |
| 表五：ANNC\_DICT（含义Announcement Dict每个公告词典信息） | | | |
| ANNC \_NO | 该公告标号 | INT UNSIGNED | 4 |
| ANNC\_DICT \_WORD | 该公告的某个词 | VARCHAR | 32 |
| ANNC\_DICT \_WNUM | 该公告的某个词的数量 | INT UNSIGNED | 4 |

下划线的属性是主码

**4.2、 数据结构设计**



**4.3、 数据存储设计**

存储长度，类型见上表格，

存储位置、保密方面不考虑

时间效率暂不考虑，如果有同学想改进，只需要先考虑把字词标号，可以大幅缩短检索速度。

# 6、系统出错处理

try:

没有返回会造成后续处理过程异常的模块

except：

可能会循环调用，如果是不可逆的错误，会返回并修改

ProcessInfo=Tk.StringVar()#进度信息

提示用户程序出问题了。只返回当前函数的函数名。

else:

继续执行

# 7、其他设计

暂无