更具体的操作，标注在了代码注释里，可以按需求查看

按文件名顺序依次执行，是一个完整流程

## 文档说明

### 1文件插入数据库

# 此文件是将文本存入数据库

# 在此之前，先要有自己的数据库文件，

# 如果没有可能，连接到我的 test.db中

# 文本文件，执行一次，会在表中插入所有文件，如果要更新表格，需要将表格 TABLE1 删除

# 路径替换为自己本机的路径



### 2关键词查找

# 直接描述和间接描述的数组

# 需要更改的话一个[]中，前面是基础词，后面是定位词

# 如第一个["disasters", "natural,nature,weather,environmental"] 中，

# "disasters"是基础词， "natural,nature,weather,environmental"是定位词

# 要加入新的，就新建一个[]，

# 更新原本的话，直接在原本的数组中，加入单词

# 将其余列内容复制到新表，注意此sql只有不存在表 自然灾害风险描述 的时候运行

# 如果已经存在表 自然灾害风险描述 ，先将它删除，再运行这个文件



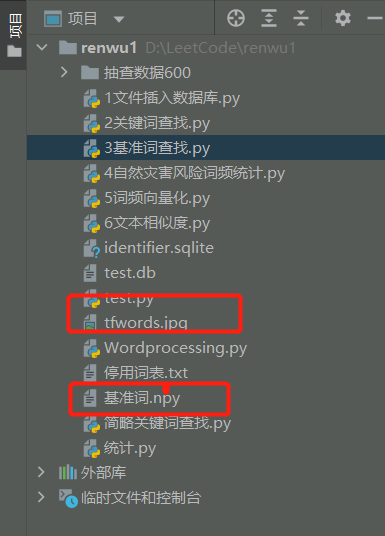
### 3基准词查找

# 从文件中查找所有风险描述中 词语出现的频率

# 用numpy保存在 基准词.py 文件中

# 可以替换停用词表.txt， 直接将自己想要去掉的词语或者符号，添加到文件中即可

# 取出了出现频率最高的20个词语，并画图显示，并将图片保存到了 tfwords.jpg





### 4风险词频统计

# 将每个文本中的描述，和前面统计的基准词，做比较，得到词频

# 此处1-600个文件，一样是根据总文件，601改为最后的文件数量+1

# 最后词频保存到了表 自然灾害风险描述 中的新加列中

### 5词语向量化

# 根据词语出现的此处，向量化

# 本文件不建议修改，处了总文件数 601-》改为你的文件数

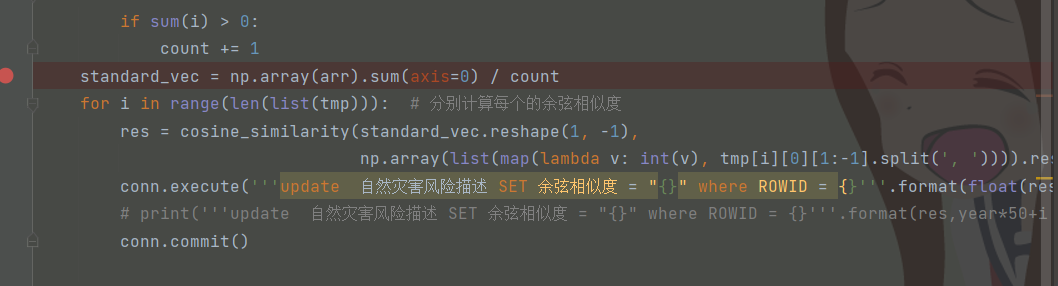


### 6文本相似度

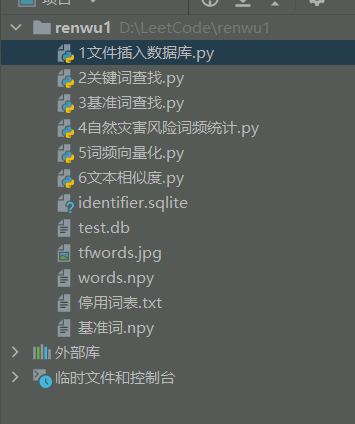
# 本文件中，将上述5文件中得到的，词语向量，用余弦相似度函数计算得到

# 其中按每年前20个公司，为基准，并去除为0的公司，求均值，得到基准向量

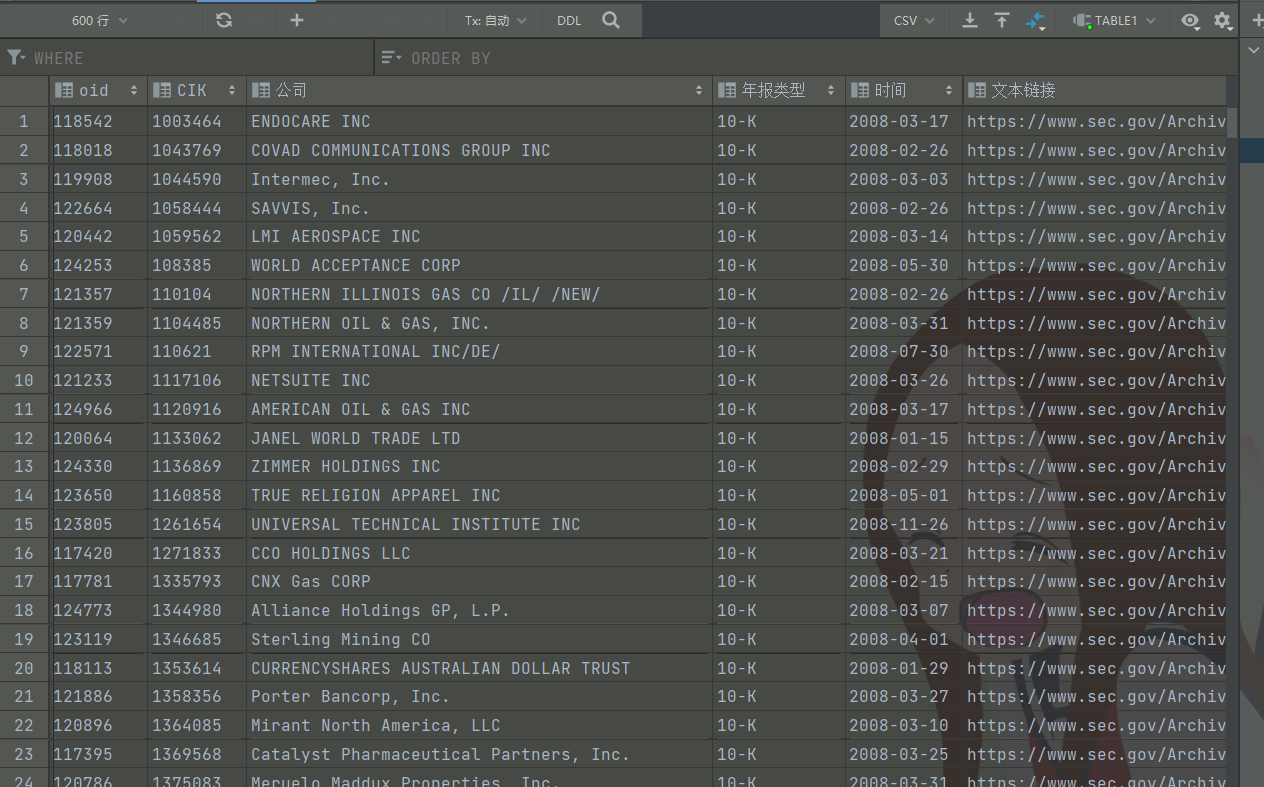
# 然后用基准向量 与每一个求余弦值



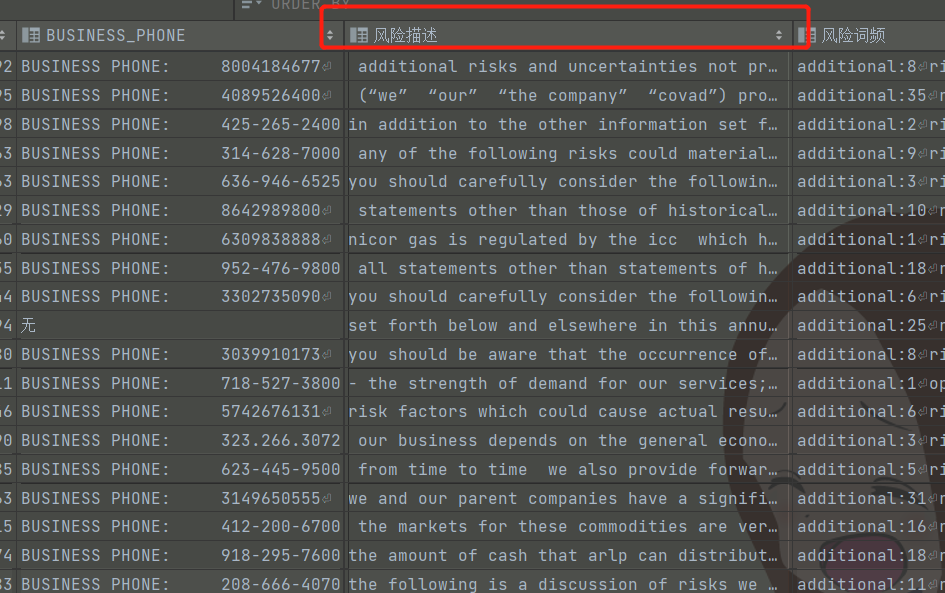
## 流程



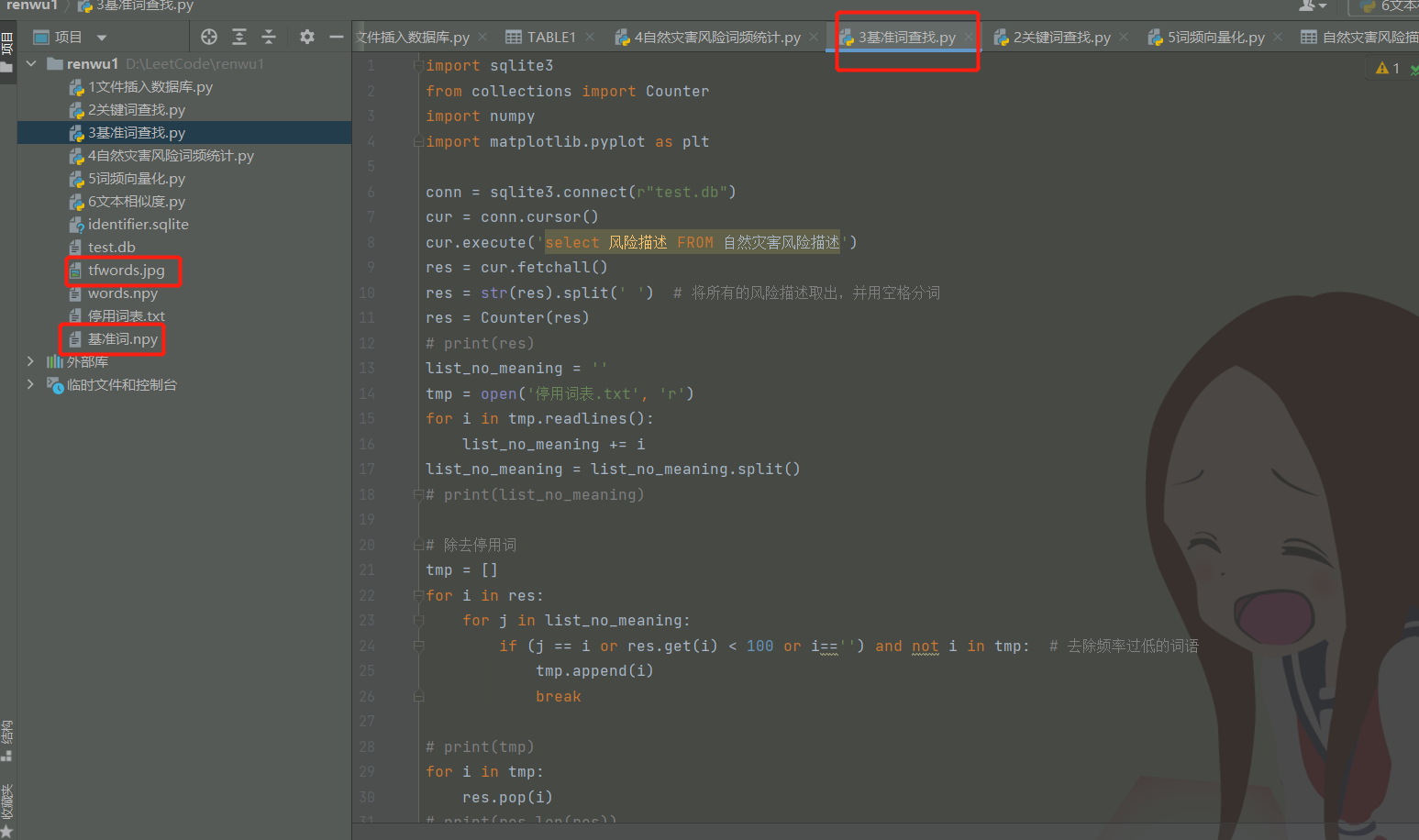
### 1、将所有数据插入数据库，得到TEBALE1

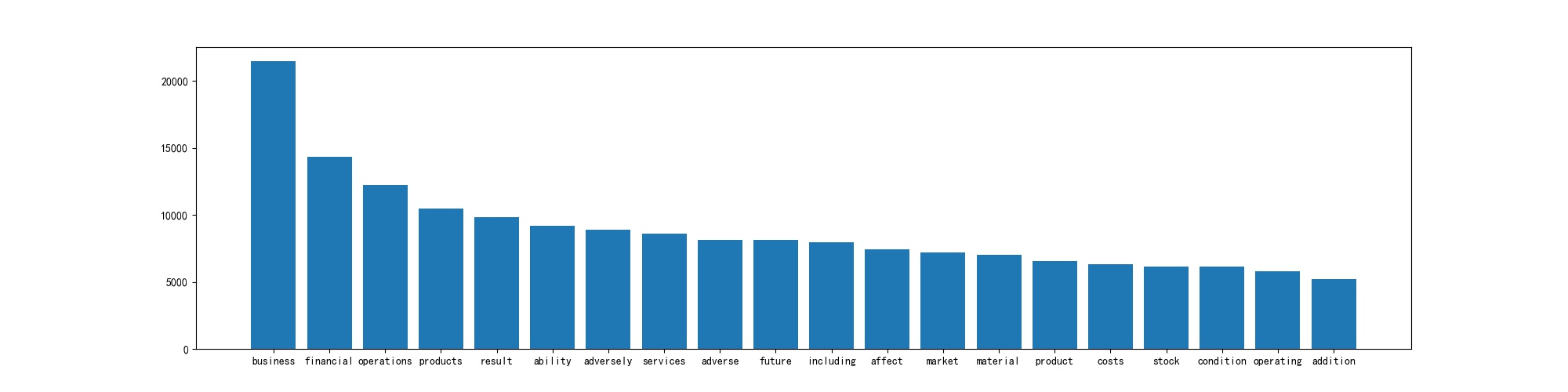


### 2、将定位词从文本中提取出来，并将其插入新建表‘自然灾害风险描述‘

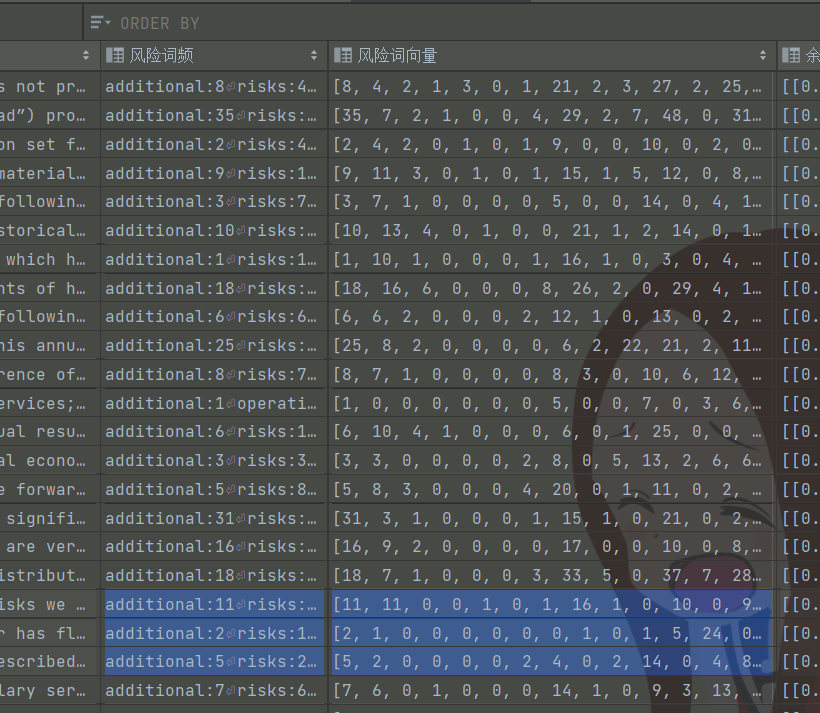


### 3、通过所有的文本中词频，统计出现较多的，并保存到基准词中，并取出数量最多的前20个画图看一下。





### 4、统计每个文本中基准词出现的次数，并存入表‘’自然灾害风险描述“（没有新建表，在后面新插入了一列），并将风险词按出现频率向量化



### 5、将每年前n（n=20）个公司的向量求均值作为基准，计算当年所有公司何其的余弦相似度。

