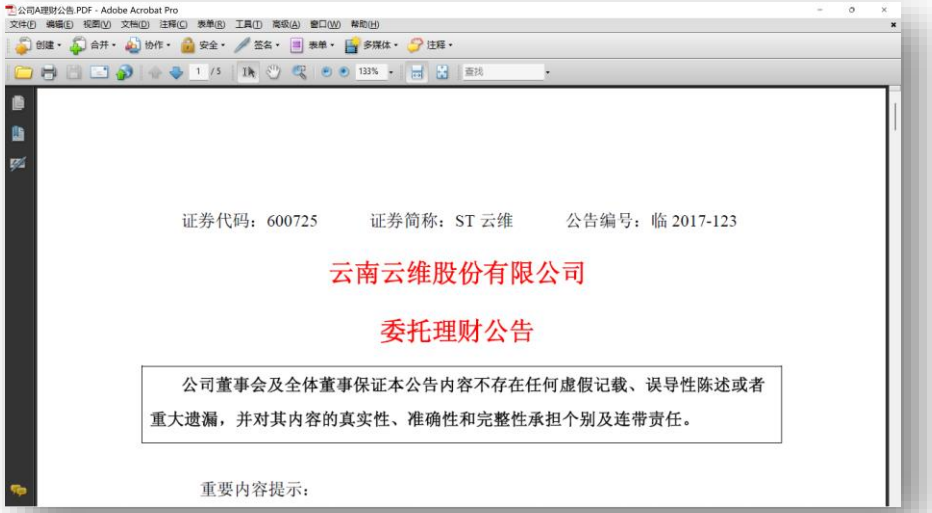
Python与金融数据挖掘(4)

文欣秀

wenxinxiu@ecust.edu.cn

案例分析



证券代码：600725　　证券简称：ST 云维　　公告编号：临 2017-123

云南云维股份有限公司

委托理财公告

公司董事会及全体董事保证本公告内容不存在任何虚假记载、误导性陈述或者重大遗漏，并对其内容的真实性、准确性和完整性承担个别及连带责任。

重要内容提示：

数据类型

Python数据类型

- 数字
  - 整型
  - 浮点型
  - 布尔型
  - 复数
- 有序序列
  - 字符串
  - 列表
  - 元组
- 无序序列
  - **集合**
  - **字典**

代码	名称
000001	上证指数
399001	深证成指
899050	北证50
000300	沪深300
399005	中小100
399006	创业板指

常用字典方法

di.keys()： 返回包含字典所有**键**的对象

di.values()： 返回包含字典所有**值**的对象

di.items()： 返回包含所有(**键**、**值**)项对象

di.get(key,[default])： 返回健**key对应的值**，若key不存在，则返回default

di.update(a)： 将字典a中的键值对添加到di中

二十大报告词云案例

二十大报告.txt - 记事本

文件　编辑　查看

同志们：

现在，我代表第十九届中央委员会向大会作报告。

中国共产党第二十次全国代表大会，是在全党全国各族人民迈上全面建设社会主义现代化国家新征程、向第二个百年奋斗目标进军的关键时刻召开的一次十分重要的大会。

大会的主题是：高举中国特色社会主义伟大旗帜，全面贯彻新时代中国特色社会主义思想，弘扬伟大建党精神，自信自强、守正创新，踔厉奋发、勇毅前行，为全面建设社会主义现代化国家、全面推进中华民族伟大复兴而团结奋斗。

关于文本词频统计

词频统计的内涵：累加问题，即对文档中的每个词设计一个计数器，词语出现一次，计算器加1，词和次数是一对出现，构成

<单词>：<出现次数>

键值对：字典

词频统计问题的IPO描述

输入 → **处理** → **输出**

从文件中读取一篇待分析的文章

采用字典数据结构统计词语出现的频率

根据词频进行图形绘制或统计高频词语

jieba库分词模式

精确模式：将句子最精确地切开，适合文本分析

```
>>>import jieba
>>>jieba. lcut("中华人民共和国是一个伟大的国家")
```

二十大报告词云案例（1）

```
import jieba
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from imageio.v2 import imread
fobj=open("二十大报告.txt","r",encoding="utf-8")
txt=fobj. read()
words=jieba. lcut(txt)
```

单词计数方法一

```
aList=["上海","北京", "上海","云南","北京", "上海"]
counts={}
for word in aList:
    if   word   not in   counts:
        counts[word]=1
    else:
        counts[word]=counts[word]+1
print(counts)
```

单词计数方法二

```
aList=["上海","北京", "上海","云南","北京", "上海"]
counts={}
for word in aList:
    counts[word]=counts. get(word,0)+1
print(counts)
```

二十大报告词云案例（2）

```
counts={}
for word in words:
        if len(word)==1:
            continue
        else:
            counts[word]=counts. get(word,0)+1
pic = imread('cloud.jpg')
```

二十大报告词云案例（3）

```
wc=WordCloud(mask=pic,font_path='msyh.ttc',  #中文字体
            repeat=False, #内容可以重复
            background_color='white', #设置背景颜色
            max_words=110,          #设置最大词数
            max_font_size=120,     #设置字体最大值
            min_font_size=10,     #设置字体最小值
            random_state=50,       #设置配色方案
            scale=10)
wc.generate_from_frequencies(counts)
plt.imshow(wc) #将数值以图片形式显示出来
plt.show()
```

二十大报告词云案例

问题： 如何去掉部分不重要的词语？

集合定义

- 集合使用大括号 **{}** 来包裹
- 集合相当于**只有键没有值**的字典
- 集合内的元素**不可重复**出现
- 集合内的元素是**不可变**的
- 集合内的元素**没有**先后关系

集合运算—示例

```
>>> a={"江西铜业","神州长城","中集集团","古井贡酒"}
>>> h={"中集集团","江西铜业","小米集团","阿里影业"}
>>> a & h          {'中集集团', '江西铜业'}
>>> a | h          {'中集集团', '古井贡酒', '小米集团', '阿里影业', '神州长城', '江西铜业'}
>>> a - h          {'神州长城', '古井贡酒'}
>>> a ^ h          {'小米集团', '阿里影业', '神州长城', '古井贡酒'}
>>> "小米集团" not in a   True
```

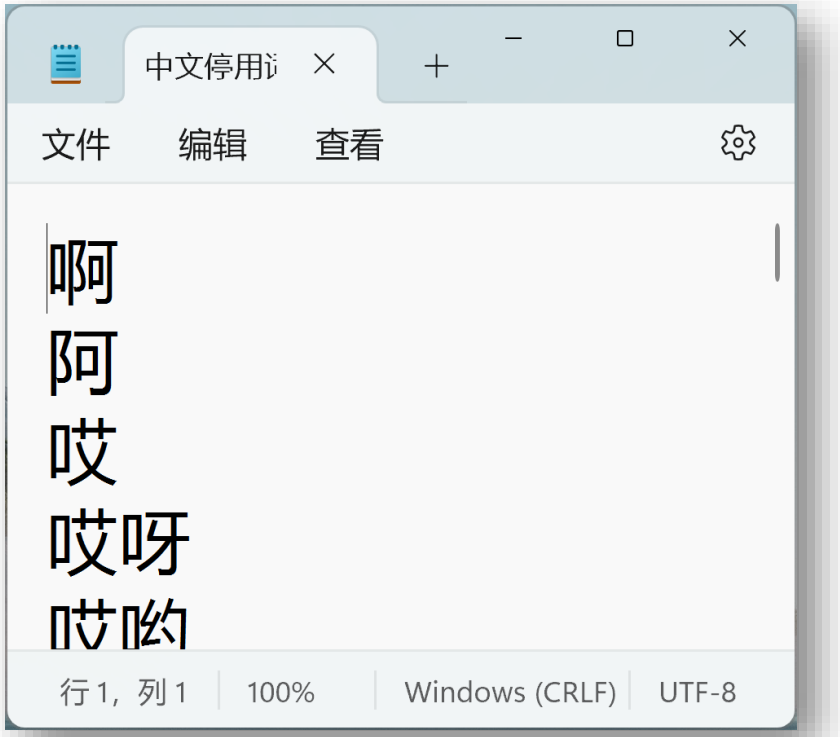
集合运算二示例

```
>>> a={"江西铜业","神州长城","中集集团","古井贡酒"}
>>> h={"中集集团","江西铜业","小米集团","阿里影业"}
>>> s={"江西铜业","中集集团"}
>>> s<=a
True
>>> s > h
False
>>> s< a
True
>>> a==h
False
```

二十大报告词云案例（修改2）

```
counts={}
excludes={"不断","一系列","基本"}
for word in words:
        if len(word)==1:
            continue
        elif word in excludes:
            continue
        else:
            counts[word]=counts. get(word,0)+1
```

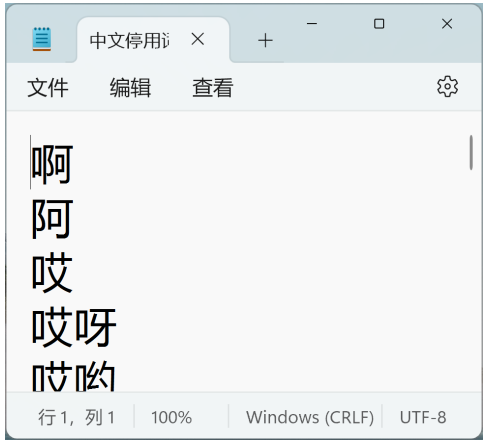
拓展问题



思考： 如何从文件中读取所有停用词并进行判断？

从文件获取排除词

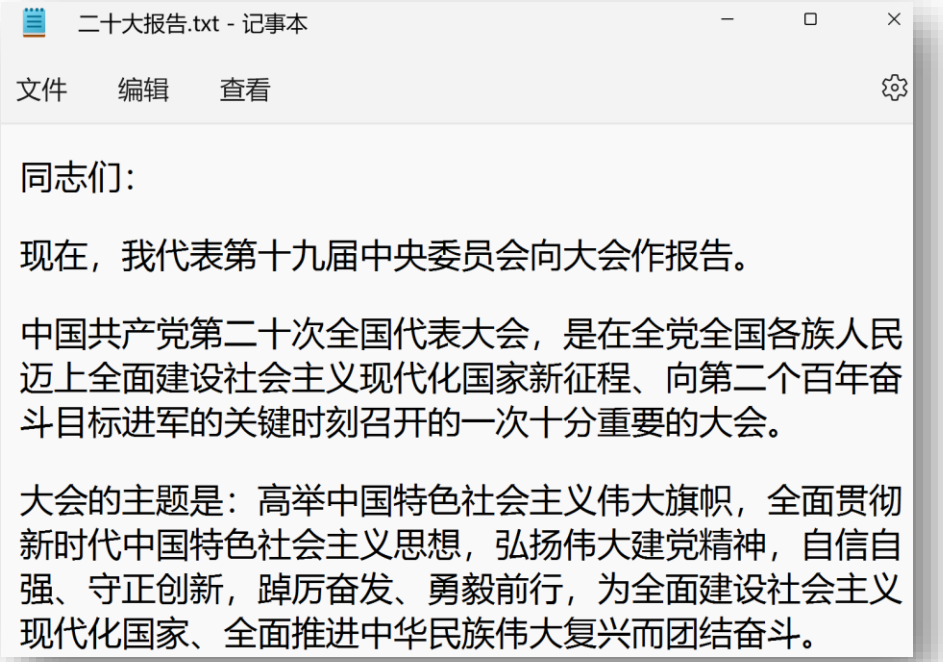
```
excludes=set()
with open("中文停用词.txt", "r") as handle:
    for i in handle:
      i=i. strip()
      excludes. add(i)
```

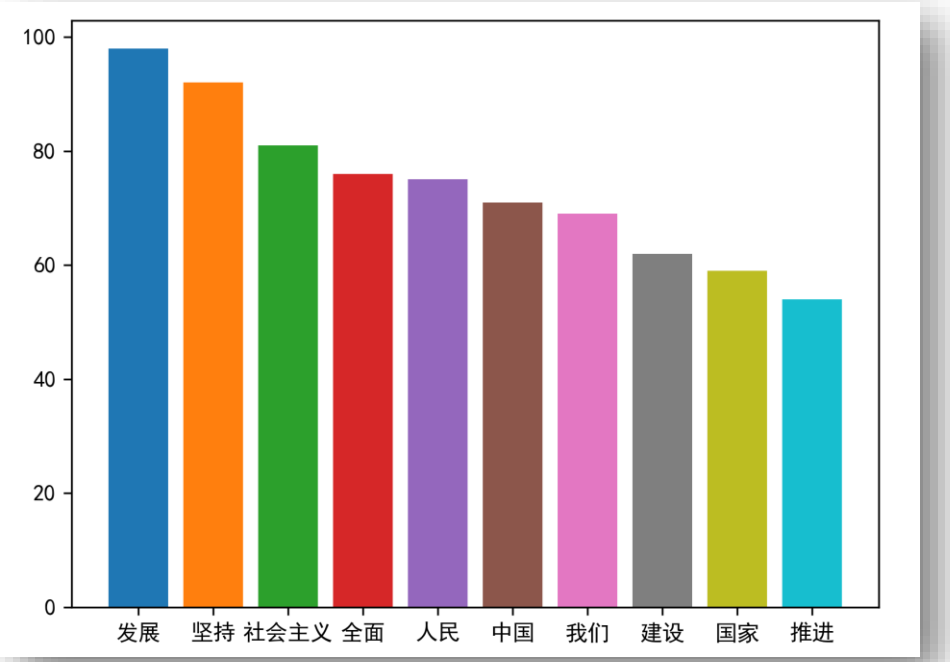


二十大报告词云案例（修改3）

```
counts={}
excludes=set()
with open("中文停用词.txt","r",encoding="utf-8") as fobj:
    for i in fobj:
        i=i. strip()
        excludes. add(i)
for word in words:
    if len(word)==1:
        continue
    elif word in excludes:
        continue
    else:
        counts[word]=counts. get(word,0)+1
```

拓展问题





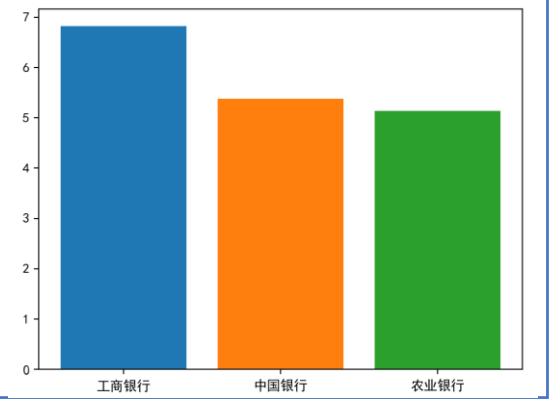
问题： 如何根据词频画出柱状图？

字典按值排序案例

```
shares = {"中国银行": 5.38, "工商银行": 6.82, "农业银行": 5.14}
items=list(shares. items())
items. sort(key=lambda x:x[1], reverse=True)
print(items)
```

基于列表绘图案例

```
import matplotlib.pyplot as plt
shares = {"中国银行": 5.38, "工商银行": 6.82, "农业银行": 5.14}
items=list(shares. items())
items. sort(key=lambda x:x[1], reverse=True)
plt.rcParams['font.sans-serif']=['SimHei']   #用来正常显示中文标签
for i in items:
    name, price=i[0], i[1]
    plt. bar(name, price)
plt. show()
```



二十大报告词频统计案例

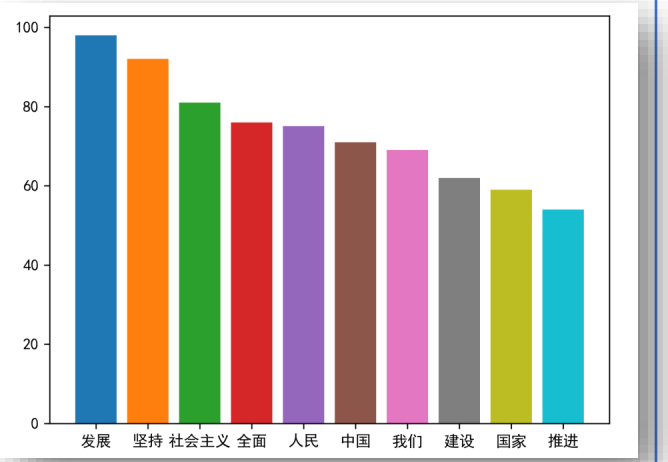
```
import jieba
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from imageio.v2 import imread
fobj=open("二十大报告.txt","r",encoding="utf-8")
txt=fobj.read()
words=jieba.lcut(txt)
```

二十大报告词频统计案例

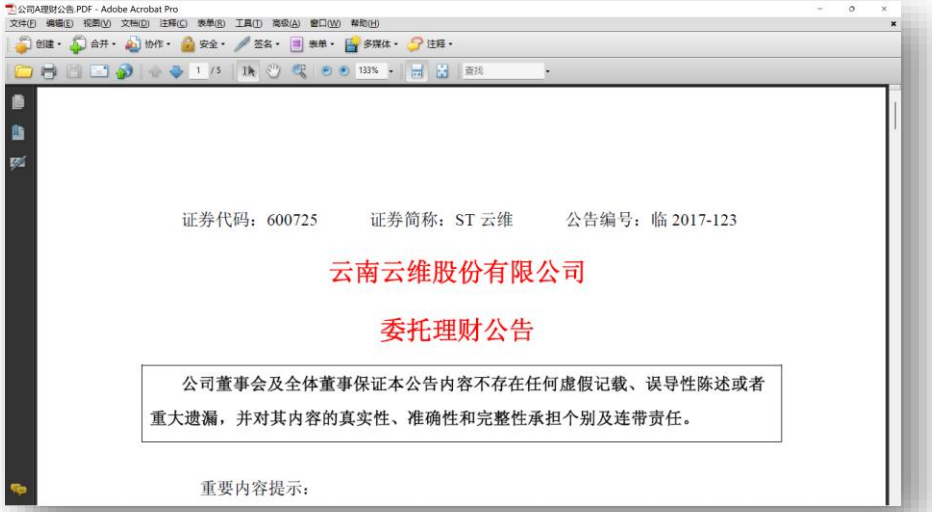
```
counts={}
excludes=set()
with open("中文停用词.txt","r",encoding="utf-8") as fobj:
    for i in fobj:
        i=i. strip()
        excludes. add(i)
for word in words:
    if len(word)==1:
        continue
    elif word in excludes:
        continue
    else:
        counts[word]=counts. get(word,0)+1
```

二十大报告词频统计案例

```
items=list(counts. items())
items. sort(key=lambda x:x[1],reverse=True)
plt.rcParams['font.sans-serif']=['SimHei']    #用来正常显示中文标签
for i in range(10):
    word, count=items[i]
    plt. bar(word,count)
plt.show()
```



拓展问题



证券代码：600725　　证券简称：ST 云维　　公告编号：临 2017-123

云南云维股份有限公司

委托理财公告

公司董事会及全体董事保证本公告内容不存在任何虚假记载、误导性陈述或者重大遗漏，并对其内容的真实性、准确性和完整性承担个别及连带责任。

重要内容提示：

思考： 如何从pdf文件中读取数据进行词频统计？

PDF文件读取

```
import pdfplumber
pdf = pdfplumber.open('公司A理财公告.PDF')
pages = pdf. pages
text_all = []
for page in pages:                    # 遍历pages中每一页的信息
    text = page. extract_text()  # 提取当页的文本内容
    text_all. append(text)          # 汇总每一页内容
text_all = ". join(text_all)        # 把列表转换成字符串
print(text_all)                     # 打印全部文本内容
pdf. close()
```

生成词典

```
import jieba
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from imageio.v2 import imread
words=jieba.lcut(text_all)
counts={}
for word in words:
        if len(word)==1:
            continue
        else:
            counts[word]=counts. get(word,0)+1
pic = imread('cloud.jpg')
```

绘制词云

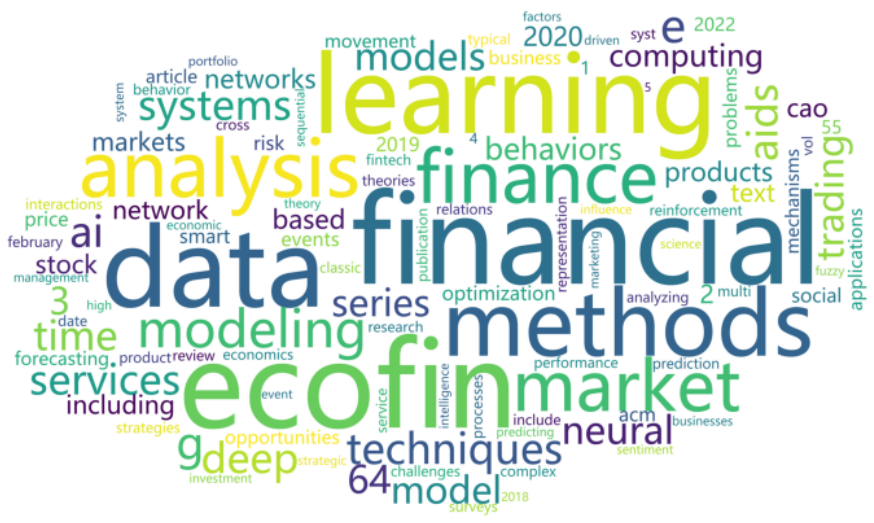
```
wc=WordCloud(mask=pic,font_path='msyh.ttc',  #中文字体
            repeat=False, #内容可以重复
            background_color='white', #设置背景颜色
            max_words=110,          #设置最大词数
            max_font_size=120,     #设置字体最大值
            min_font_size=10,     #设置字体最小值
            random_state=50,       #设置配色方案
            scale=10)
wc.generate_from_frequencies(counts)
plt.imshow(wc) #将数值以图片形式显示出来
plt.show()
```

思　考

AI in Finance: Challenges, Techniques, and Opportunities

LONGBING CAO, University of Technology Sydney, Australia

AI in finance refers to the applications of AI techniques in financial businesses. This area has attracted attention for decades, with both classic and modern AI techniques applied to increasingly broader areas of finance, economy, and society. In contrast to reviews on discussing the problems, aspects, and opportunities of finance benefited from specific or some new-generation AI and data science (AIDS) techniques or the progress of applying specific techniques to resolving certain financial problems, this review offers a comprehensive and dense landscape of the overwhelming challenges, techniques, and opportunities of AIDS research in finance over the past decades. The challenges of financial businesses and data are first outlined, followed by a comprehensive categorization and a dense overview of the decades of AIDS research in finance. We then structure and illustrate the data-driven analytics and learning of financial businesses and data. A comparison, criticism, and discussion of classic versus modern AIDS techniques for finance follows. Finally, the open issues and opportunities to address future AIDS-empowered finance and finance-motivated AIDS research are discussed.



解题步骤

第一步：转换英文文献为文本文件，保存为AI_in_Finance.txt

读取文件内容存入字符串中

```
fobj = open("AI_in_Finance.txt", "r")
paper=fobj. read()
```

解题步骤

第二步：分解并提取英文文章的单词

- 通过paper. lower()函数统一字母为小写
- 使用paper. replace()方法将英文单词的分隔符(空格、标点符号或者特殊符号）统一为空格
- 使用paper. split()方法分解单词

字符串常量（string模块）

常量名称	常量内容
string.punctuation	'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
string.digits	'0123456789'
string. ascii_letters	'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
string.printable	'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&\'()*+,./:;<=>?@[\\]^_`{|}~\t\n\r\x0b\x0c'
string.whitespace	' \t\n\r\x0b\x0c'

代码实现（一）

```
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from imageio.v2 import imread
import string
fobj = open("AI_in_Finance.txt", "r")
paper=fobj. read()
paper = paper. lower()
for ch in string.punctuation:
    paper = paper. replace(ch, " ") #将特殊字符替换为空格
words = paper.split( )
```

问题分析

第三步：统计每个单词出现次数：全部单词保存在列表words 中，定义一个字典counts={}，键值对为：

<单词>：<出现次数>

依次统计每个单词出现次数

单词计数方法

```
if   word   in   counts:
      counts[word]=counts[word]+1
else:
      counts[word]=1
```

可简化为:

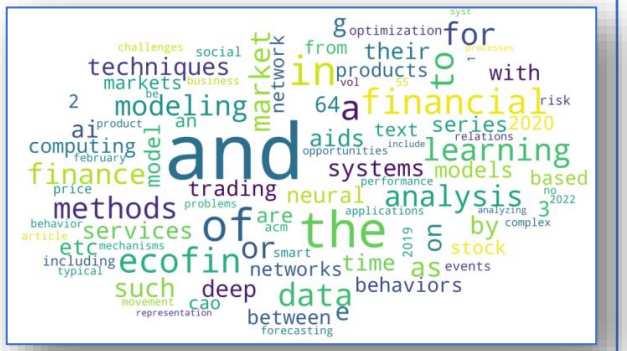
```
counts[word]=counts. get(word,0)+1
```

代码实现（二）

```
counts = { }
for word in words:
    counts[word] = counts. get(word,0) + 1
print(counts)
```

代码实现（三）

```
pic = imread('cloud.jpg')
wc=WordCloud(mask=pic,font_path='msyh.ttc',  #中文字体
            repeat=False, background_color='white', #设置背景颜色
            max_words=110, max_font_size=120,    #设置字体最大值
            min_font_size=10, random_state=50,     #设置配色方案
            scale=10)
wc.generate_from_frequencies(counts)
plt.imshow(wc) #将数值以图片形式显示出来
plt.show()
```



问　题

分析结果表明高频词汇都是**冠词**、**代词**、**连接**词等语法型词汇，并不能代表文章含义。

可以采用**集合**类型构建一个排除词汇库**excludes**，将语法型词汇排除。

去除无意义词

excludes={"and", "of", "a", "for", "in", "on", "the", "to"}

……

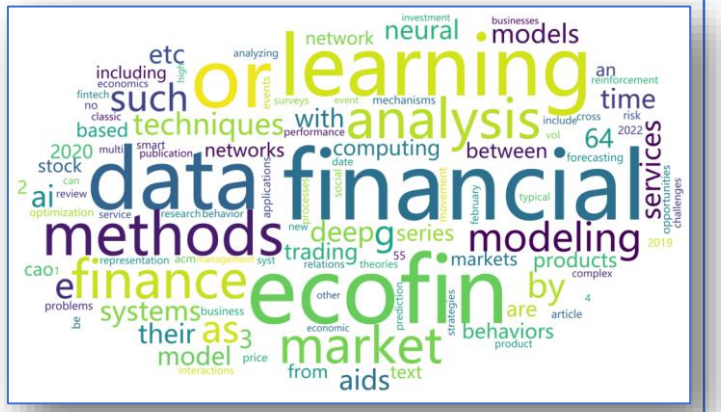
if word in excludes:

continue

……

代码实现（二）修改

```
counts = { }
excludes={"and", "of", "a", "for",  "in", "on","the","to"}
for word in words:
    if word in excludes:
        continue
    counts[word] = counts. get(word,0) + 1
print(counts)
```



拓展练习

```
able
about
above
according
accordingly
across
actually
after
afterwards
again
against
ain't
all
allow
allows
```

如何从文件中读取所有停用词并进行判断?

从文件获取排除词

```
excludes=set()
with open("stop.txt", "r") as handle:
    for i in handle:
      i=i. strip()
      excludes. add(i)
```

stop.txt - 记事本

文件 编辑 查看

```
able
about
above
according
accordingly
across
actually
after
afterwards
again
```

代码实现（二）修改

```
excludes=set()
with open("stop.txt", "r") as handle:
    for i in handle:
      i=i. strip()
      excludes. add(i)
counts = { }
for word in words:
    if word in excludes:
        continue
    else:
        counts[word] = counts. get(word,0) + 1
```

函 数 定 义

```
def 函数名 ([形式参数列表]):
    执行语句
    [return 返回值]
```

```
def mul(x,y):
    z=x*y
    return z
```

```
num1=int(input("请输入第一个数："))
num2=int(input("请输入第二个数："))
result=mul(num1,num2)
print("结果是%d" % result)
```

函 数 定 义

```
def  函数名 ([形式参数列表]):
        执行语句
        [return 返回值]
```

```
def draw():
    for i in range(100):
        circle(i)
        right(90)
```

```
from turtle import *
speed(0)
pencolor("red")
draw()
```

参 数 传 递

- 位置传递
- 关键字传递
- 默认值参数传递
- 元组传递
- 字典传递

位置传递

- 调用函数时，**按照函数声明时参数顺序**依次进行参数传递

```
def fun1(a, b):
    if (a>b):
        return a
    else:
        return b

print(fun1(7, 3))
```

7

关键字传递

- 调用函数时，**明确指定把某个实参值**传递给某个形参

```
def fun2(a, b):
    if (a>b):
        print("a=",a)
        return a
    else:
        return b

print(fun2(b=2,a=7))
```

```
a= 7
7
```

默认值参数传递

- 在定义函数时**直接对形参赋值**

```
def fun3(a, b=2):
    if (a>b):
        return a
    else:
        return b

print(fun3(7))
```

7

默认值参数传递

- 函数调用时，可以**全部、部分或不用**默认值

```
def area(r=1.0, pi=3.14):
    return r*r*pi

print("面积1=%.2f" % area())     #全部用默认值
print("面积2=%.2f" % area(5.1))  #部分用默认值
print("面积3=%.2f" % area(5.1,3.14159)) #不用默认值
```

```
面积1=3.14
面积2=81.67
面积3=81.71
```

元组类型变长参数传递

元组类型变长参数传递：在函数声明时，若在某个参数名称前面加一个星号“*”，则表示该参数是一个元组类型可变长参数。

```
def mul(*number):
    print("元素为：",number)
    total=1
    for i in number:
        total=total*i
    return total
```

```
print("结果为：" , mul(2,4))
print("结果为：" , mul(1,2,3,4,5))
```

```
File  Edit  Shell  Debug  Options  Window  Help
元素为：  (2, 4)
结果为：  8
元素为：  (1, 2, 3, 4, 5)
结果为：  120
>>>
Ln: 9  Col: 4
```

字典类型变长参数传递

字典类型变长参数传递：在函数声明时，若在其某个参数名称前面加两个星号“**”，则表示该参数是一个字典类型可变长参数。

```
def func(**dict):
    print(dict)
    print(sum(dict.values()))

func(a=1,b=2,c=3)
```

```
File  Edit  Shell  Debug  Options  Window  Help
{'a': 1, 'b': 2, 'c': 3}
6
>>>
```

lambda匿名函数

lambda： Python预留的关键字，可以定义一个匿名函数，函数体是一个简单的表达式而不是语句块，通常用在只使用一次的场景

形　式： lambda argument_list: expression

例　子： **lambda x: x**3** 输入x，输出x的3次方

lambda匿名函数

- 可以把匿名函数赋值给一个变量，再调用该函数

```
def f(x,y):
    return x+y
x,y=2,3
print("x+y=%d"%(f(x,y)))
```

```
x,y=2,3
z=lambda x,y:x+y
print("x+y=%d"%(z(x,y)))
```

map( ) 函数

- 接收一个函数 f 和一个或多个list，并通过把函数 f 依次作用在list 的每个元素上，得到一个新的 list 并返回。

```
>>> price=[35.8, 24.9, 23.5]
>>> list(map(lambda x: x*2, price))
```

```
>>> price1=[35.8, 24.9, 23.5]
>>> price2=[88.5, 32.0, 12.9]
>>> list(map(lambda x,y: x+y, price1, price2))
```

谢　谢