

光大证券 金融工程部	文档性质	软件版本	文档版本
	程序技术文档	--	v1.0

关键词词频&网络图 程序

技术文档

文档作者： _____ 罗 剑_____

日期：_2012_/_8_/_28_



版权所有 不得复制

一、功能概述

关键词词频&网络图是以股票论坛、个股新闻、研究报告三个网站作为数据源，以文本数据挖掘作为核心技术，以 Lucene 检索作为系统框架，以证券分析为目的，实现的智能文本分析系统，该系统主要实现了以下功能：

- 关键词词频统计
- 关键词网络图

其中，关键词词频统计功能是：对于给定的关键词(Word)以及给定的股票代码(Ticker)在一定的时间范围[StartDate, EndDate]内，计算每周的平均词频占比，同时给出该词频占比时间序列与股价之间的相关系数。关键词词频统计的结果实例如图 1-1 所示：

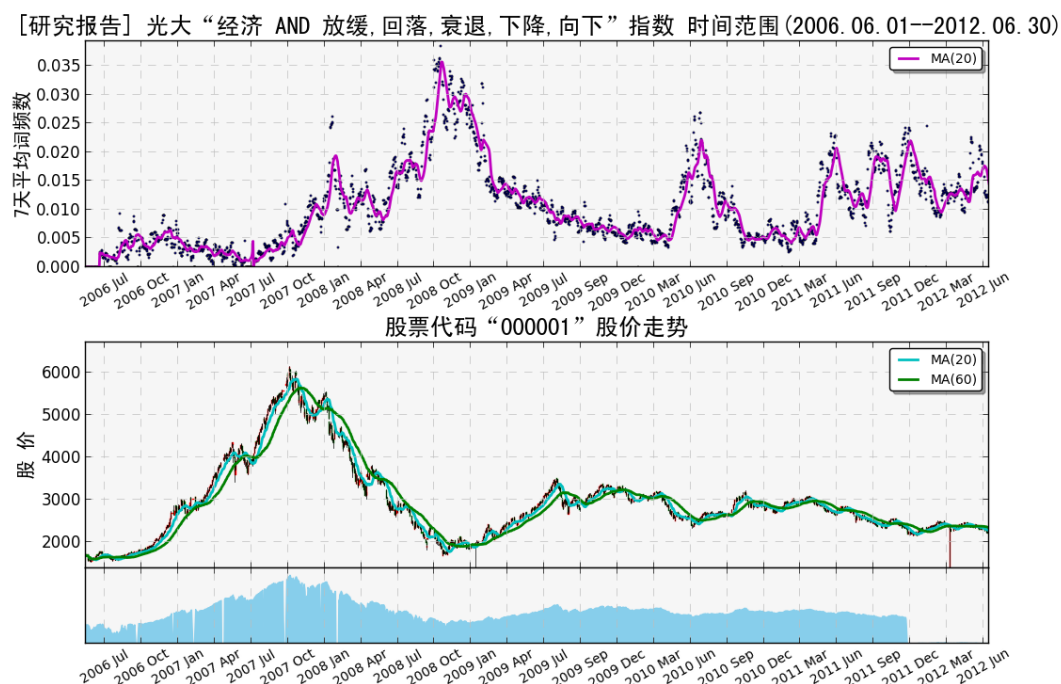


图 1-1 关键词词频统计功能结果示意图

关键词网络图的功能是：对于给定的关键词(Word)在一定的时间范围[StartDate, EndDate]内，根据 TF-IDF 关联度指标为依据，给出与关键词最相关

的 20 个一级词，以及与一级词最相关的 5 个二级词，组成关键词网络图。关键词网络图的结果实例如图 1-2 所示：

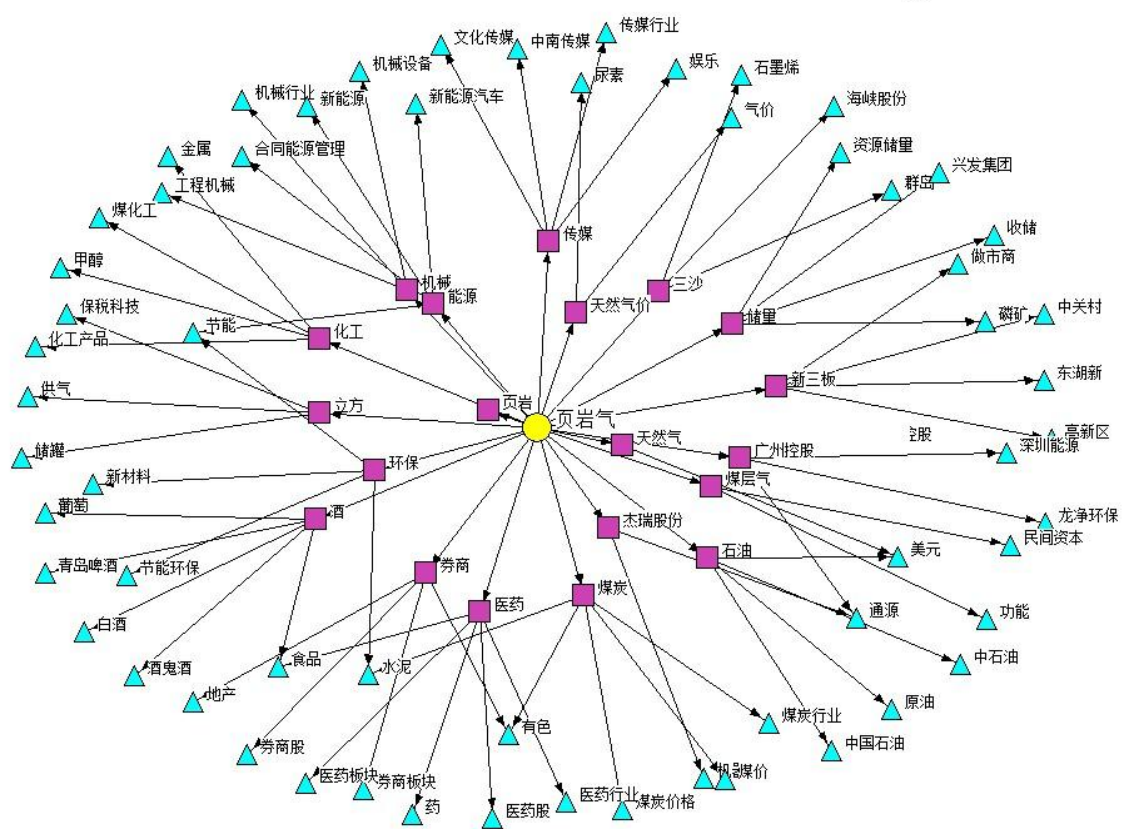


图 1-2 关键词网络结果示意图

二、功能模块

如本文之前所述，本文分析系统的两个功能是建立在三个文本的数据源，Lucene 检索的架构之上的。所以，实现关键词词频、关键词网络图的功能需要先做一些准备工作和模块支持，本文所实现的智能文本分析系统只要分为以下几个模块：

- 爬虫模块
- 检索模块
- 统计模块
- 关键词词频模块
- 关键词网络模块

整个系统的功能模块构成如图 2-1 所示:

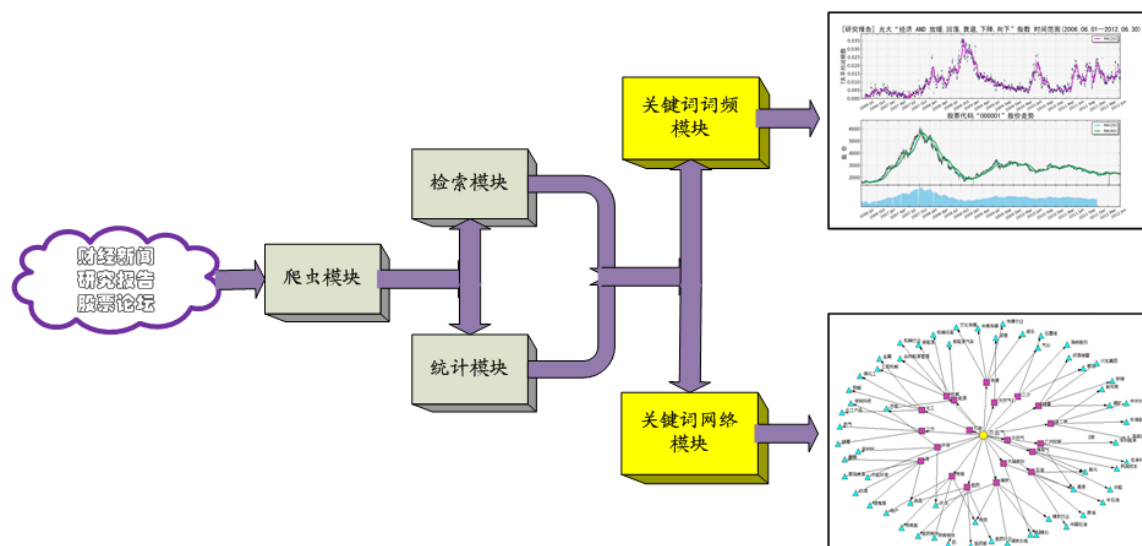


图 2-1 关键词词频&网络图功能模块构成图

通过功能模块构成图，可以看出，这五个功能模块中，爬虫模块、统计模块、检索模块为三个基础模块，构成了智能文本分析系统的基础模块；关键词词频模块、关键词网络模块是架在三个基础模块之上的应用模块，用来实现特点的词频统计和产生网络图的功能。

本文先简单阐述一下每个模块的功能和作用，每个模块具体的运行方法将在随后的章节中给予详细地介绍。

1) 爬虫模块

爬虫模块的主要作用在于将股票论坛、个股新闻、研究报告三个网站的网页数据通过网页解析的方式将文本内容爬下来，用于之后模块的文本挖掘。爬虫模块将爬到的文本数据以【日期 + 股票代码】为单位存至相应的 TXT 文本文件当中，同时将文本文件所在的位置以及其他相关信息写入数据库。

对于每个数据源，都有一个独立的程序进行网页爬虫，他们分别是：

- [GetGuba_pylucene.py](#) 股票论坛网页爬虫
- [GetMbReport_pylucene.py](#) 研究报告网页爬虫
- [GetSinaNews_pylucene.py](#) 个股新闻网页爬虫

2) 检索模块

检索模块的主要作用在于以 Lucene 为架构，将爬虫模块爬到的文本数据加入到全文索引当中，在建立索引的过程中，系统以“句子”作为基本的检索单位，

即检索关键词能够定位到该关键词所在的句子。另外，索引采用增量的方式来建立，即每次只将最新爬的文本加入到搜索索引当中，而对于三个数据源，系统分别建立了三个独立的索引。同时，在建立的索引的基础上，检索模块还实现了基本的文本检索功能，检索程序能够在一定的时间范围内对于检索给定关键词，并返回该关键词所在的存储文件的文件名，以及该关键词所在的“句子”，并将所有的检索结果输出到一个给定的文件中。

简而言之，检索模块提供了建立索引和文本搜索两个主要的功能，他们分别是：

- *IndexFiles_pylucene.py* 增量建立索引
- *SearchFiles_pylucene.py* 关键词全文检索

3) 统计模块

设计统计模块是为了随后的关键词词频和网络模块进行数据的准备，和爬虫模块和检索模块一样，统计模块也是基础模块。统计模块的主要功能有三个：

- 以【用户字典】为列表，计算用户字典中每个关键词在三个数据源中出现的总词频数
- 以【用户字典】为列表，计算用户字典中每个关键词在三个数据源中出现过的总文档数
- 以句子为单位，计算三个数据源中每天文档的总句子数

其中，用户字典关键词的词频数和文档数，是为了关键词网络模块中计算TF-IDF 相关度指标所准备的数据，而每天的句子数则树为了关键词词频模块中计算词频占比所准备的数据。

对于统计模块的这三个功能，分别有三个独立的程序进行，他们分别是：

- *IDFCalWord.py* 计算关键词总词频数
- *IDFCal.py* 计算关键词所在文档数
- *SentenceCal.py* 计算每天文档的句子数

4) 关键词词频模块

通过建立三个基础模块，能够完成一系列的应用，关键词词频模块是其中的一个应用模块，关键词词频模块的主要功能在于：对于给定的关键词以及给定的

股票代码，在一定的时间范围内，计算每周的平均词频占比，给出词频占比序列的曲线和股票价格曲线的对比图，同时给出该词频占比与股价之间的相关系数。

模块中没有直接使用关键词每天的词频，而是根据每天的词频，以及当天文档的句子总数计算关键词的词频占比。对于关键词 x ，词频占比 $index_x$ 的计算公式如下：

$$index_x = \frac{\sum(N_x^i)}{\sum(Sen^i)} = \frac{\sum_{i=1}^7 N_x^i}{\sum_{i=1}^7 Sen^i}$$

其中，公式的各个指标的意义如下：

- N_x^i : 概念关键词 x 在第 i 天出现的次数
- Sen^i : 第 i 天中文档的句子总数

从词频占比的计算公式可以看出，词频占比是将每个星期的关键词的词频总和除以每个星期文档的句子总数得到的。使用词频占比而非直接采用词频，能够更公平地反应出关键词 x 每天的关注程度，从而更合理地对词频信号进行使用。另外，在计算关键词与给定股票的相关系数时，模块会以一周为频率计算关键词的词频占比时间序列，同时计算该周内给定股票股价的均值，计算两个时间序列的相关系数作为两者相关性的依据。

对于关键词词频模块，只有一种调用的方式，调用时需给出关键词、股票代码以及时间范围：

- `sigWordSeq.py` 关键词词频时间序列

5) 关键词网络模块

和关键词词频模块一样，关键词网络模块也属于应用模块，关键词网络模块的主要功能在于：对于给定的关键词、在一定的时间范围内，根据 TF-IDF 关联度指标为依据，给出与关键词最相关的 20 个一级词，以及与一级词最相关的 5 个二级词，组成关键词网络图。

其中关联度指标采用的是TF-IDF算法，TF-IDF是一种常用的文本检索与本文探勘的加权技术，主要用于评估某个词对于一份特定文档的重要程度。在本文的关键词网络模块中，将给定关键词的搜索结果集合作为特定文档，TF-IDF用于评估搜索结果中每个词对于该结果的关联程度，即对于关键词的关联程度。TF-IDF的具体计算公式如下：

$$x_{tf_idf} = x_{tf} * \log\left(\frac{1}{x_{df}}\right) = \frac{n_x}{n_{all}} * \log\left(\frac{N_{all}}{N_x}\right)$$

其中 x 为搜索结果中的某个词， x_{tf_idf} 为词 x 的与关键词的TF-IDF关联度指标，其他符号意义如下：

- n_x : 词 x 在概念关键词检索结果中出现的次数
- n_{all} : 概念关键词检索结果中出现的总词数
- N_x : 词 x 在数据源所有文本数据中出现的次数
- N_{all} : 数据源所有文本数据的文件数
- x_{tf} : 词 x 在概念关键词检索结果中出现的词频
- x_{df} : 词 x 在数据源所有文本数据中出现的词频

其中 x 的词频参数 x_{tf} 和 x_{df} 可以由其他参数计算得到：

- $x_{tf} = \frac{n_x}{n_{all}}$
- $x_{df} = \frac{N_x}{N_{all}}$

对于关键词网络模块，提供了两种形式的调用，一是对于给定的关键词，生成完整的关键词网络图，二是对于只给出与关键词关联度最高的20只股票组合，他们分别是：

- `WordNet.py` 完整关键词网络图
- `WordNet_stock.py` 关键词关联股票组合

三、模块运行

如之前所述，本文的智能文本分析系统的五个组成模块中，爬虫模块、统计模块、检索模块为三个基础模块，关键词词频模块、关键词网络模块是两个依附于基础模块的应用模块。在本小节中，本文将详细阐述每个模块的输入参数、运行方式、以及运行步骤和结果。

1) 爬虫模块

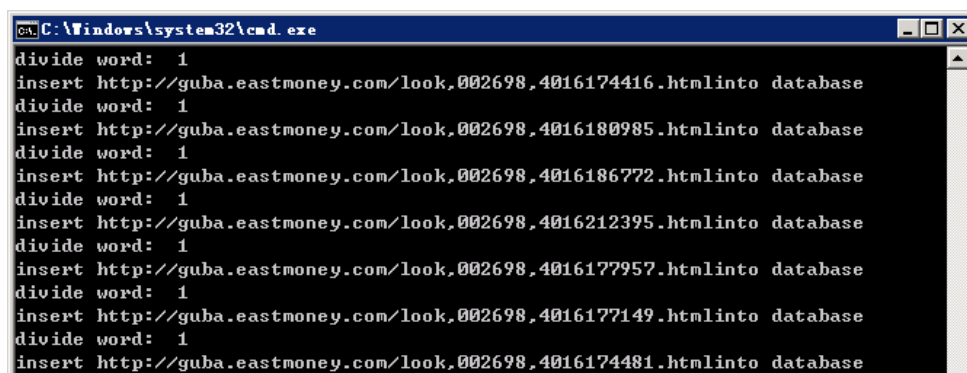
路径：`D:\TotalCode\LuceneCode\GetData\GetGuba_pylucene.py`

功能：股票论坛网页爬虫

输入参数：无

运行举例：`python GetGuba_pylucene.py`

运行过程实例：



```
C:\Windows\system32\cmd.exe
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016174416.htmlinto database
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016180985.htmlinto database
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016186772.htmlinto database
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016212395.htmlinto database
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016177957.htmlinto database
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016177149.htmlinto database
divide word: 1
insert http://guba.eastmoney.com/look,002698,4016174481.htmlinto database
```

图 3-1 股票论坛网页爬虫运行过程

运行结果：

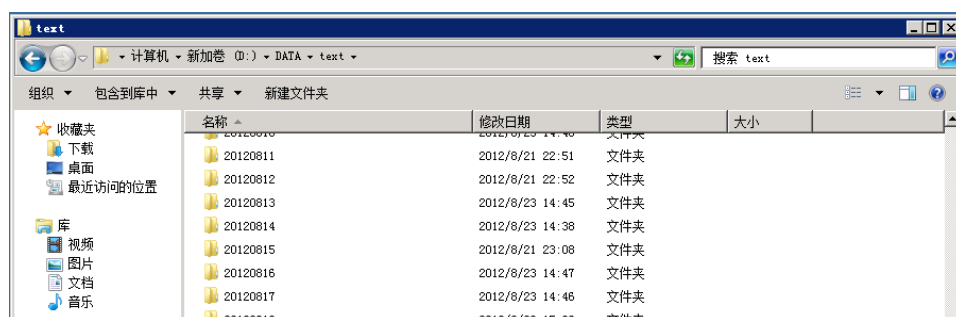


图 3-2 股票论坛网页爬虫运行结果

如图 3-2 所示，股票论坛网页爬虫程序会将爬得的网页文本数据按日期分别存放至文件夹【D:\DATA\text】中，而相应的分好词的数据存放至文件夹【D:\DATA\Lucene\text】。

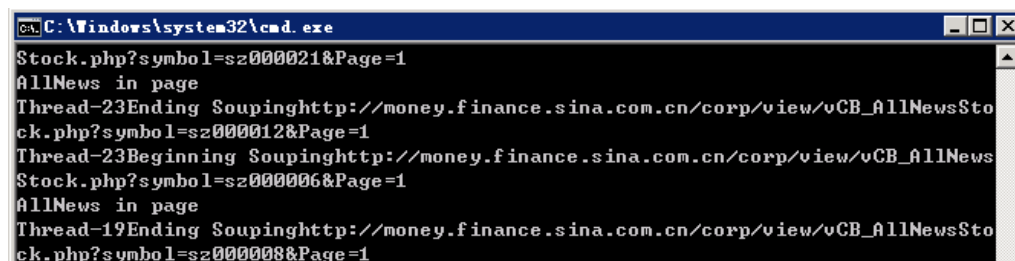
路径：[D:\TotalCode\LuceneCode\GetData\GetSinaNews_pylucene.py](#)

功能：个股新闻网页爬虫

输入参数：无

运行举例： `python GetSinaNews_pylucene.py`

运行过程实例：



```
C:\Windows\system32\cmd.exe
Stock.php?symbol=sz000021&Page=1
AllNews in page
Thread-23Ending Soupinghttp://money.finance.sina.com.cn/corp/view/vCB_AllNewsSto
ck.php?symbol=sz000012&Page=1
Thread-23Beginning Soupinghttp://money.finance.sina.com.cn/corp/view/vCB_AllNews
Stock.php?symbol=sz000006&Page=1
AllNews in page
Thread-19Ending Soupinghttp://money.finance.sina.com.cn/corp/view/vCB_AllNewsSto
ck.php?symbol=sz000008&Page=1
```

图 3-3 个股新闻网页爬虫运行过程

运行结果：

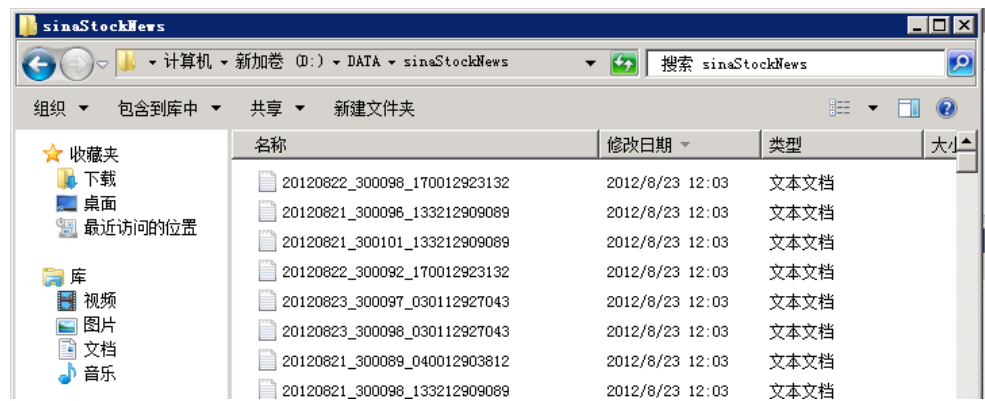


图 3-4 个股新闻网页爬虫运行结果

如图 3-4 所示，个股新闻网页爬虫程序会将爬得的网页文本数据按【日期_股票代码_新闻编号】进行命名，并存放至文件夹【D:\DATA\sinaStockNews】中，而相应的分好词的数据存放至文件夹【D:\DATA\Lucene\sinaStockNews】。

路径：[D:\TotalCode\LuceneCode\GetData\GetMbReport_pylucene.py](#)

功能：研究报告网页爬虫

输入参数：无

运行举例：`python GetMbReport_pylucene.py`

运行过程实例：

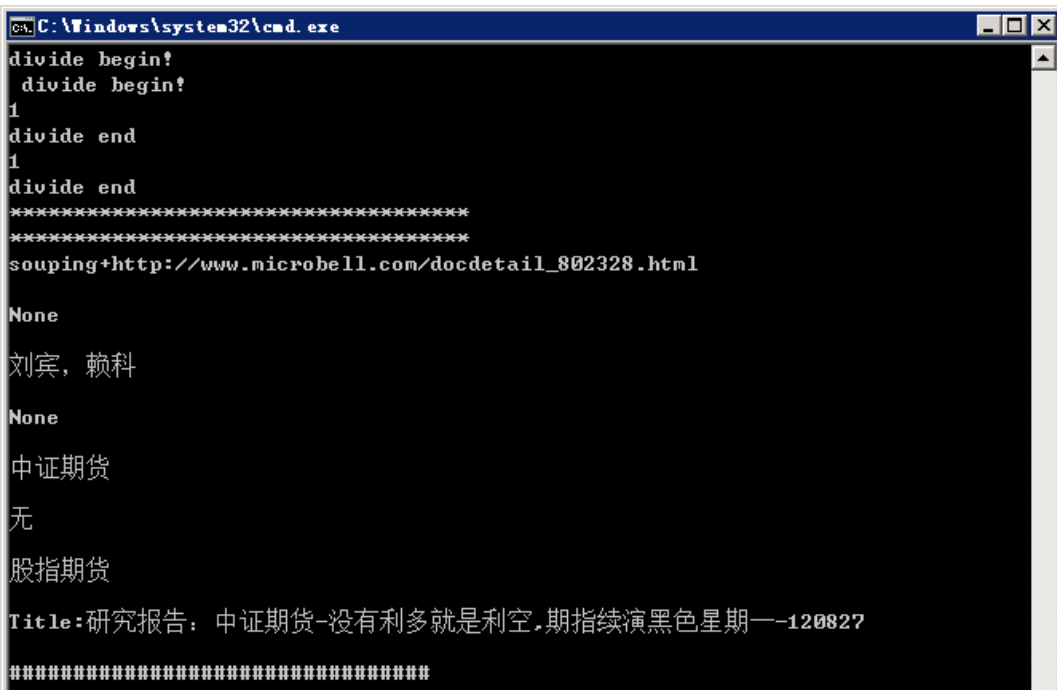


图 3-5 研究报告网页爬虫运行过程

运行结果：

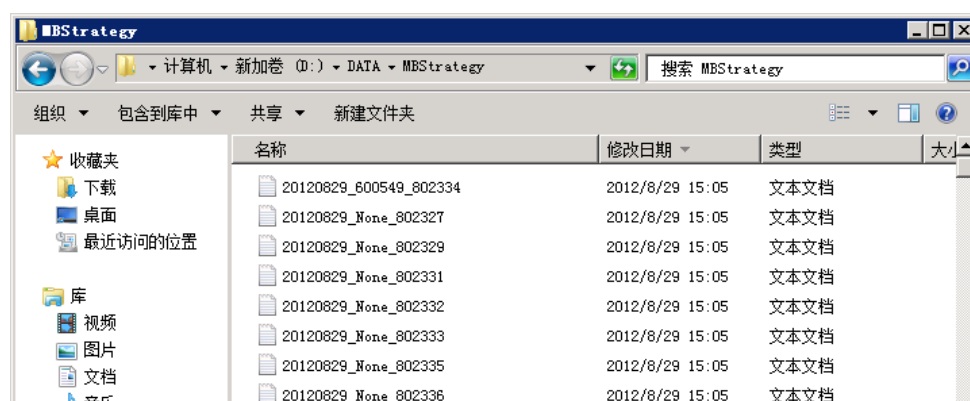


图 3-6 研究报告网页爬虫运行结果

如图 3-6 所示，研究报告网页爬虫程序同样会将爬得的网页文本数据按【日期_股票代码_报告编号】进行命名，并存放至文件夹【D:\DATA\MBStrategy】中，而相应的分好词的数据存放至文件夹【D:\DATA\Lucene\MBStrategy】。

2) 检索模块

路径：[D:\TotalCode\LuceneCode\Index_Search\IndexFiles_pylucene.py](#)

功能：增量建立索引

输入参数：〈数据目录〉〈索引目录〉〈开始日期〉〈结束日期〉

运行举例： `python IndexFiles_pylucene.py D:\DATA\text D:\DATA\Index\text 20120715 20120820`

运行过程实例：

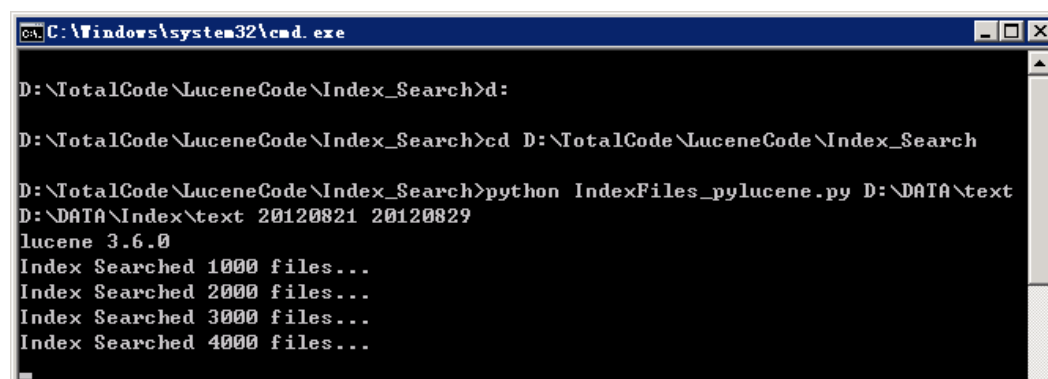


图 3-7 增量建立索引运行过程

运行结果：

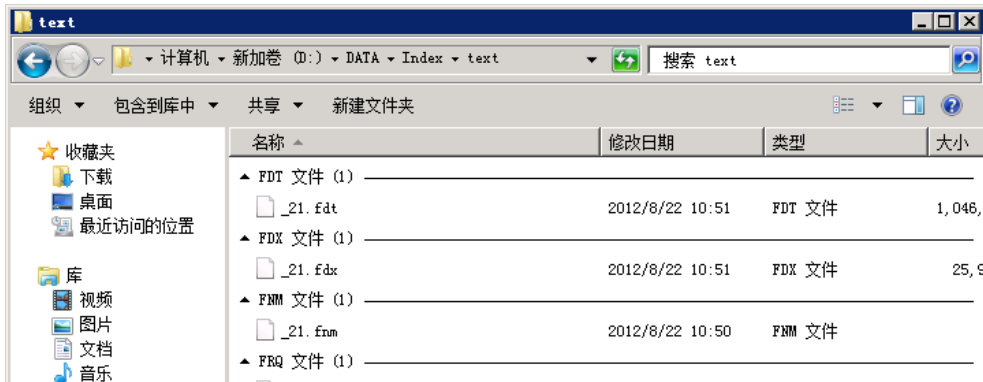


图 3-8 增量建立索引运行结果

如图 3-8 所示，增量建立索引程序会扫描所有数据目录下的文件，并将在日期属于【开始日期，结束日期】的文件数据，增加到索引当中。

路径：[D:\TotalCode\LuceneCode\Index_Search\SearchFiles_pylucene.py](#)

功能：关键词全文检索

输入参数：<索引目录> <关键词> <输出文件>

运行举例：`python SearchFiles_pylucene.py D:\DATA\Index\text "页岩气"`

`D:\TotalCode\LuceneCode\Index_Search\Output_pylucene.txt`

运行过程实例：

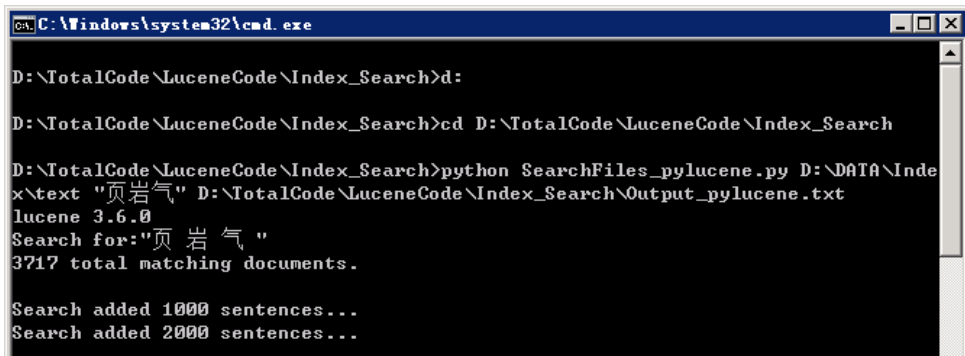


图 3-9 关键词全文检索运行过程

运行结果：

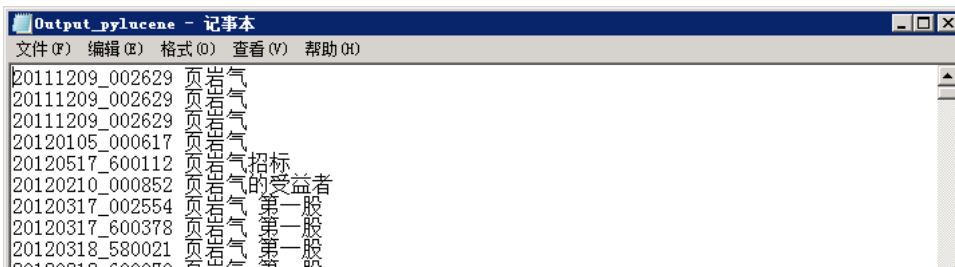


图 3-10 关键词全文检索运行结果

如图 3-10 所示，关键词全文检索程序会在给定的索引中寻找所有包含关键词的句子，并将所有搜索结果按【文件名_句子】的形式存放在输出文件中。

3) 统计模块

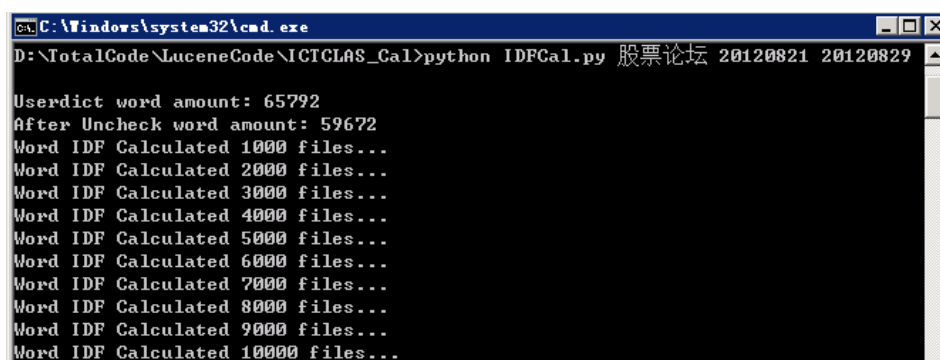
路径：[D:\TotalCode\LuceneCode\ICTCLAS_Cal\IDFCal.py](#)

功能：计算关键词所在文档数

输入参数：〈数据源〉〈开始日期〉〈结束日期〉

运行举例：`python IDFCal.py 股票论坛 20120715 20120820`

运行过程实例：

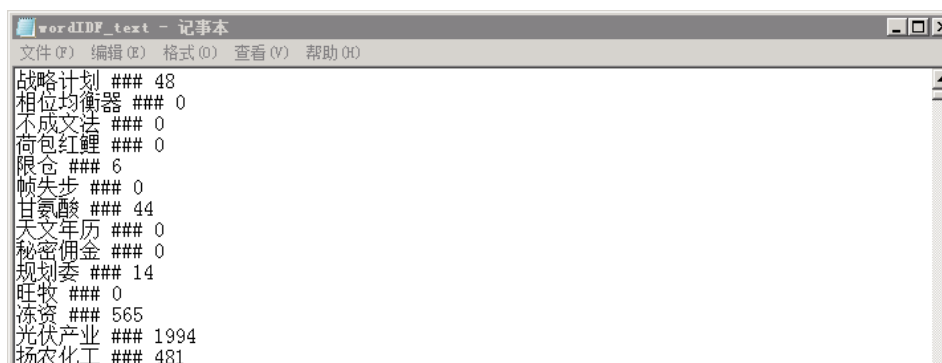


```
C:\Windows\system32\cmd.exe
D:\TotalCode\LuceneCode\ICTCLAS_Cal>python IDFCal.py 股票论坛 20120821 20120829

Userdict word amount: 65792
After Uncheck word amount: 59672
Word IDF Calculated 1000 files...
Word IDF Calculated 2000 files...
Word IDF Calculated 3000 files...
Word IDF Calculated 4000 files...
Word IDF Calculated 5000 files...
Word IDF Calculated 6000 files...
Word IDF Calculated 7000 files...
Word IDF Calculated 8000 files...
Word IDF Calculated 9000 files...
Word IDF Calculated 10000 files...
```

图 3-11 计算关键词所在文档数运行过程

运行结果：



```
wordIDF_text - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

战略规划 ### 48
相位均衡器 ### 0
不成文法 ### 0
荷包红鲤 ### 0
限仓 ### 6
帧失步 ### 0
甘氨酸 ### 44
天文年历 ### 0
秘密佣金 ### 0
规划委 ### 14
旺牧 ### 0
冻资 ### 565
光伏产业 ### 1994
扬农化工 ### 481
```

图 3-12 计算关键词所在文档数运行结果

如图 3-12 所示，计算关键词所在文档数的程序会在给定的数据源中寻找所有日期在时间范围【开始日期，结束日期】内的文档，计算文档内所有词出现过的文档数，并增量更新至文件【D:\ICTCLAS\wordIDF_】中。

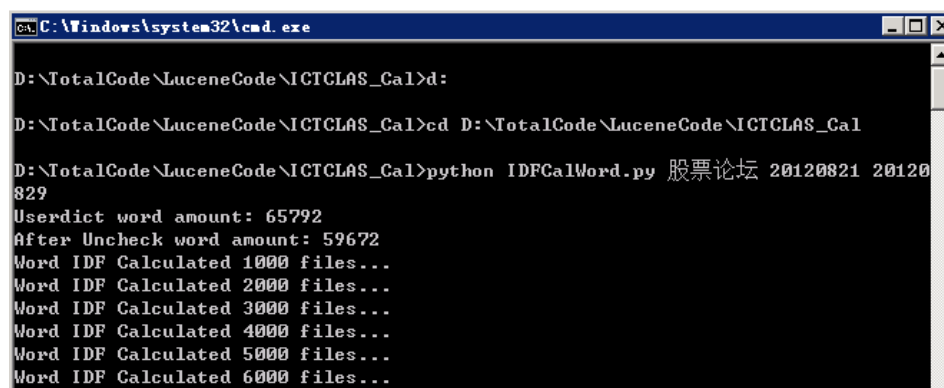
路径：[D:\TotalCode\LuceneCode\ICTCLAS_IDF\CalWord.py](#)

功能：计算关键词总词频数

输入参数：〈数据源〉〈开始日期〉〈结束日期〉

运行举例： `python IDFCalWord.py 股票论坛 20120715 20120820`

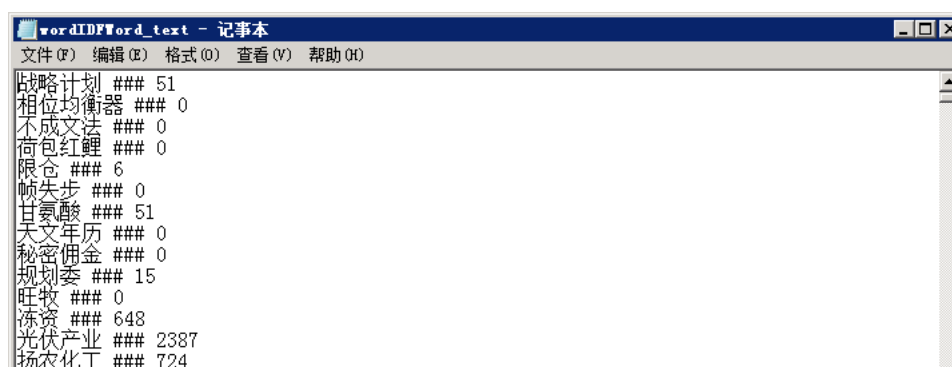
运行过程实例：



```
C:\Windows\system32\cmd.exe
D:\TotalCode\LuceneCode\ICTCLAS_Cal>d:
D:\TotalCode\LuceneCode\ICTCLAS_Cal>cd D:\TotalCode\LuceneCode\ICTCLAS_Cal
D:\TotalCode\LuceneCode\ICTCLAS_Cal>python IDFCalWord.py 股票论坛 20120821 20120829
Userdict word amount: 65792
After Uncheck word amount: 59672
Word IDF Calculated 1000 files...
Word IDF Calculated 2000 files...
Word IDF Calculated 3000 files...
Word IDF Calculated 4000 files...
Word IDF Calculated 5000 files...
Word IDF Calculated 6000 files...
```

图 3-13 计算关键词总词频数运行过程

运行结果：



```
wordIDFWord_text - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
战略计划 ### 51
相位均衡器 ### 0
不成文法 ### 0
荷包红鲤 ### 0
限仓 ### 6
帧失步 ### 0
甘氨酸 ### 51
天文年历 ### 0
秘密佣金 ### 0
规划委 ### 15
旺牧 ### 0
冻资 ### 648
光伏产业 ### 2387
扬农化工 ### 724
```

图 3-14 计算关键词总词频数运行结果

如图 3-14 所示，计算关键词总词频数的程序会在给定的数据源中寻找所有日期在时间范围【开始日期，结束日期】内的文档，计算文档内所有词出现的次数，并增量更新至文件【D:\ICTCLAS\wordIDFWord_】中。

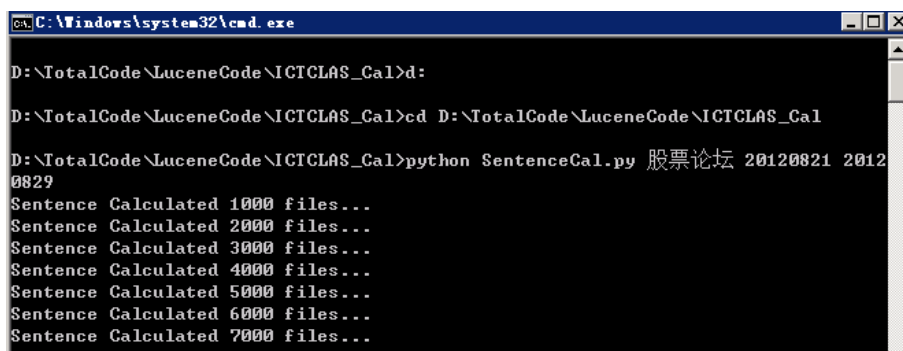
路径：`D:\TotalCode\LuceneCode\ICTCLAS_IDF\SentenceCal.py`

功能：计算每天文档的句子数

输入参数：〈数据源〉〈开始日期〉〈结束日期〉

运行举例： `python SentenceCal.py 股票论坛 20120715 20120820`

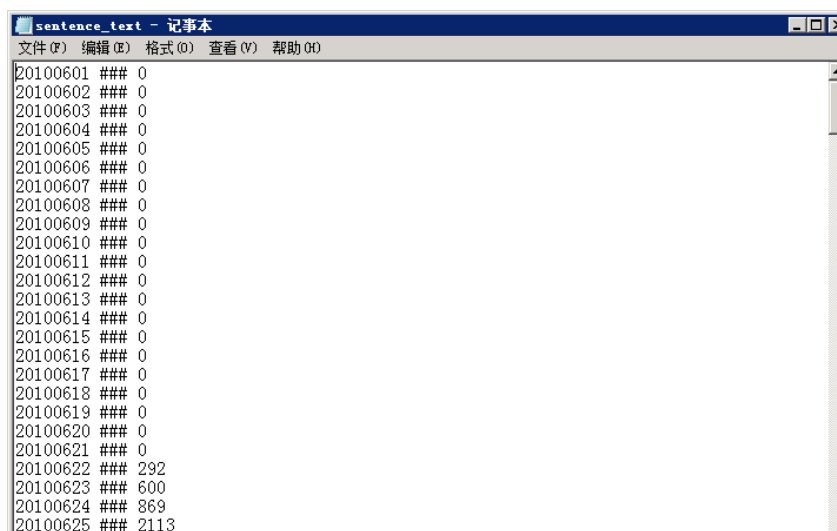
运行过程实例：



```
GA: C:\Windows\system32\cmd.exe
D:\TotalCode\LuceneCode\ICTCLAS_Cal>d:
D:\TotalCode\LuceneCode\ICTCLAS_Cal>cd D:\TotalCode\LuceneCode\ICTCLAS_Cal
D:\TotalCode\LuceneCode\ICTCLAS_Cal>python SentenceCal.py 股票论坛 20120821 2012
0829
Sentence Calculated 1000 files...
Sentence Calculated 2000 files...
Sentence Calculated 3000 files...
Sentence Calculated 4000 files...
Sentence Calculated 5000 files...
Sentence Calculated 6000 files...
Sentence Calculated 7000 files...
```

图 3-15 计算每天文档的句子数运行过程

运行结果：



```
Sentence_text - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
20100601 ### 0
20100602 ### 0
20100603 ### 0
20100604 ### 0
20100605 ### 0
20100606 ### 0
20100607 ### 0
20100608 ### 0
20100609 ### 0
20100610 ### 0
20100611 ### 0
20100612 ### 0
20100613 ### 0
20100614 ### 0
20100615 ### 0
20100616 ### 0
20100617 ### 0
20100618 ### 0
20100619 ### 0
20100620 ### 0
20100621 ### 0
20100622 ### 292
20100623 ### 600
20100624 ### 869
20100625 ### 2113
```

图 3-16 计算每天文档的句子数运行结果

如图 3-16 所示，计算每天文档句子数的程序会在给定的数据源中寻找所有日期在时间范围【开始日期，结束日期】内的文档，以天为单位计算每天文档数据的句子数，并增量更新至文件【D:\ICTCLAS\wordIDFWord_】中。

4) 关键词词频模块

路径：[D:\TotalCode\LuceneCode\ICTCLAS_IDF\sigWordSeq.py](#)

功能：关键词词频时间序列

输入参数：〈数据源〉〈关键词〉〈股票代码〉〈开始日期〉〈结束日期〉〈是否搜索标识〉

运行举例：`python sigWordSeq.py 股票论坛 "物联网" 000001 20100601 20120820 1`

运行过程实例：


```
C:\Windows\system32\cmd.exe
D:\TotalCode\LuceneCode\WordSeq>d:
D:\TotalCode\LuceneCode\WordSeq>cd D:\TotalCode\LuceneCode\WordSeq
D:\TotalCode\LuceneCode\WordSeq>python sigWordSeq.py 股票论坛 "页岩气" 000001 20
100601 20120820 1
Lucene 3.6.0
Lucene Search Init Done...
Search for:"页岩气"
3717 total matching documents.

Search added 1000 sentences...
Search added 2000 sentences...
Search added 3000 sentences...
***** Word 页岩气 Search Done *****
WordSeq Count Done...
```

图 3-17 关键词词频时间序列运行过程

运行结果:

```
C:\Windows\system32\cmd.exe
D:\TotalCode\LuceneCode\WordSeq>d:
D:\TotalCode\LuceneCode\WordSeq>cd D:\TotalCode\LuceneCode\WordSeq
D:\TotalCode\LuceneCode\WordSeq>python sigWordSeq.py 股票论坛 "页岩气" 000001 20
100601 20120820 1
Lucene 3.6.0
Lucene Search Init Done...
Search for:"页岩气"
3717 total matching documents.

Search added 1000 sentences...
Search added 2000 sentences...
Search added 3000 sentences...
***** Word 页岩气 Search Done *****
WordSeq Count Done...
0阶相关系数: -0.20770131511
1阶相关系数: -0.30419449396
2阶相关系数: -0.379073970091
3阶相关系数: -0.318297311537
4阶相关系数: -0.237266398704
请按任意键继续. . .
```

图 3-18 关键词词频时间序列运行结果

如图 3-18 所示, 关键词词频程序会在给定的数据源中计算日期在时间范围【开始日期, 结束日期】内的词频时间序列, 并计算给定股票每周平均股价和每周平均词频占比之间的相关系数, 结果输出在终端以及图片展示如图 1-1 所示。

5) 关键词网络模块

路径: `D:\TotalCode\LuceneCode\ICTCLAS_IDF\WordNet.py`

功能: 完整关键词网络图

输入参数: <数据源> <关键词> <开始日期> <结束日期>

运行举例: `python WordNet.py 研究报告 "页岩气" 20120601 20120817`

运行过程实例:

```
C:\Windows\system32\cmd.exe - python WordNet.py 研究报告 "页岩气" 20120601 20120817
D:\TotalCode\LuceneCode\WordNet>python WordNet.py 研究报告 "页岩气" 20120601 20120817
Lucene 3.6.0
Lucene Search Init Done...
start lic check
Divide Word Init Done...

Search for:"页岩气"
1758 total matching documents.

Search added 1000 sentences...
Userdict word amount: 65792
After Uncheck word amount: 59672
44436 Word Count Done...
Add Stock Code To wordDic Done...
***** 页岩气: *****
页岩[6153.26830849]
天然气[643.319129533]
煤层气[482.443762849]
能源[398.754814927]
煤炭[313.216691471]
杰瑞股份[258.014464289]
酒[240.593745742]
石油[214.873901521]
立方[198.87070434]
医药[169.232664415]
广州控股[150.714535875]
储量[136.922723213]
化工[131.490052757]
机械[128.182949196]
三沙[124.222124778]
新三板[123.331087706]
环保[110.9664702]
天然气价[110.638799739]
券商[110.29391372]
传媒[108.960128088]
```

图 3-19 关键词网络图运行过程

运行结果:

```
##### Print Word Net For String: 页岩气 #####
***** 页岩[6153.26830849]: *****
天然气 煤层气 能源 煤炭 杰瑞股份
***** 天然气[643.319129533]: *****
页岩 石油 能源 天然气价 美元
***** 煤层气[482.443762849]: *****
页岩 天然气 机械 民间资本 功能
***** 能源[398.754814927]: *****
新能源 新能源汽车 节能 页岩 合同能源管理
***** 煤炭[313.216691471]: *****
煤价 煤炭价格 煤炭行业 有色 水泥
***** 杰瑞股份[258.014464289]: *****
页岩 天然气 通源 石油 机器人
***** 酒[240.593745742]: *****
白酒 葡萄酒 酒鬼酒 青岛啤酒 食品
***** 石油[214.873901521]: *****
中石油 原油 天然气 美元 中国石油
***** 立方[198.87070434]: *****
天然气 页岩 储罐 供气 保税科技
***** 医药[169.232664415]: *****
药 食品 医药板块 医药行业 医药股
***** 广州控股[150.714535875]: *****
龙净环保 页岩 通源 杰瑞股份 深圳能源
***** 储量[136.922723213]: *****
磷矿 资源储量 兴发集团 页岩 收储
***** 化工[131.490052757]: *****
煤化工 化工产品 金属 甲醇 能源
***** 机械[128.182949196]: *****
工程机械 机械行业 机械设备 医药 煤炭
***** 三沙[124.222124778]: *****
群岛 海峡股份 新三板 页岩 石墨烯
***** 新三板[123.331087706]: *****
券商 高新区 中关村 做市商 东湖新
***** 环保[110.9664702]: *****
节能环保 医药 节能 新材料 水泥
***** 天然气价[110.638799739]: *****
天然气 页岩 尿素 气价 能源
***** 券商[110.29391372]: *****
券商股 券商板块 地产 新三板 有色
***** 传媒[108.960128088]: *****
文化传媒 传媒行业 娱乐 中南传媒 医药
请按任意键继续. . .
```

图 3-20 关键词网络图运行结果

如图 3-20 所示，关键词网络图程序会在给定的数据源中与给定关键词最相关的 20 个一级词，与一级词最相关的 5 个二级词。结果输出在终端以及图片展示如图 1-2 所示。

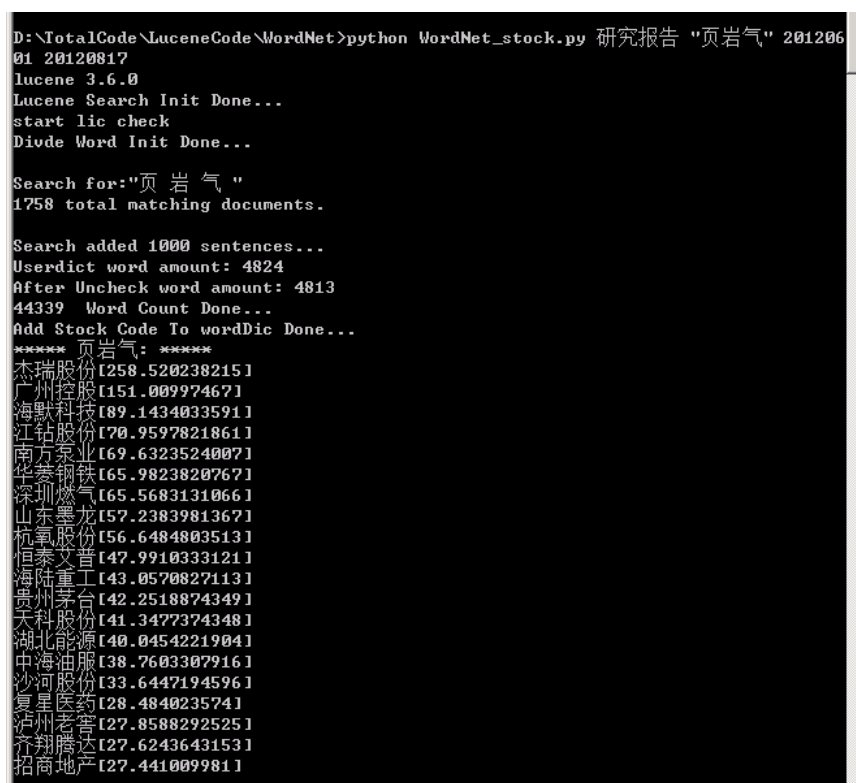
路径：[D:\TotalCode\LuceneCode\ICTCLAS_IDF\WordNet_stock.py](#)

功能：关键词关联股票组合

输入参数：〈数据源〉〈关键词〉〈开始日期〉〈结束日期〉

运行举例：`python WordNet_stock.py 研究报告 "页岩气" 20120601 20120817`

运行过程&结果：



```
D:\TotalCode\LuceneCode\WordNet>python WordNet_stock.py 研究报告 "页岩气" 20120601 20120817
01 20120817
lucene 3.6.0
Lucene Search Init Done...
start lic check
Divide Word Init Done...

Search for:"页岩气"
1758 total matching documents.

Search added 1000 sentences...
Userdict word amount: 4824
After Uncheck word amount: 4813
44339 Word Count Done...
Add Stock Code To wordDic Done...

***** 页岩气: *****
杰瑞股份 [258.520238215]
广州控股 [151.00997467]
海默科技 [89.1434033591]
江钻股份 [70.9597821861]
南方泵业 [69.6323524007]
华泰钢铁 [65.9823820767]
深圳燃气 [65.5683131066]
山东墨龙 [57.2383981367]
杭氧股份 [56.6484803513]
恒泰艾普 [47.9910333121]
海陆重工 [43.0570827113]
贵州茅台 [42.2518874349]
天科股份 [41.3477374348]
湖北能源 [40.0454221904]
中海油服 [38.7603307916]
沙河股份 [33.6447194596]
复星医药 [28.484023574]
泸州老窖 [27.8588292525]
齐翔腾达 [27.6243643153]
招商地产 [27.441009981]
```

图 3-21 关键词关联股票组合运行过程&结果

如图 3-21 所示，关键词网络图程序同样会在给定的数据源中与给定关键词最相关的 20 个词，不过这 20 个词都是股票名称。结果输出在终端上如上图所示。

四、效率性能

对于数据挖掘系统，除了模块功能本身所发挥的作用之外，还有一个非常重要的指标需要考虑，那就是模块运行的时间效率。在本小节中，本文着重对于基

基础模块中的检索模块，应用模块中的关键词词频模块、关键词网络模块的效率性能进行说明，并且用实际的实验数据来评估每个模块的时间效率。

1) 检索模块 SearchFiles

检索模块中 SearchFiles 的主要功能在于对于给定数据源的关键词进行搜索，该模块的效率性能将直接影响架构在此基础模块之上的应用模块的时间效率。本文随机抽取了 10 个关键词，计算每个关键词 SearchFiles 的累计运行时间，测试结果如图 4-1 所示：

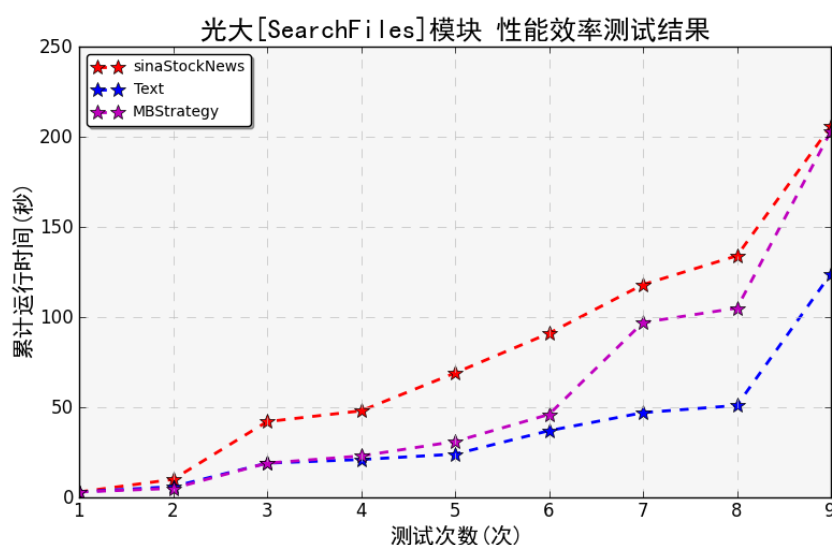


图 4-1 检索模块 SearchFiles 性能效率测试结果图

从图中可以看出，数据源个股新闻的平均耗时最高，平均每次运行时间约为 25 秒左右，数据源研究报告的平均耗时次之，平均每次运行时间约为 20 秒左右，数据源股票论坛的平均耗时是最低的，平均每次运行时间约为 15 秒左右。

2) 关键词词频模块 sigWordSeq

关键词词频模块中 sigWordSeq 的主要功能在于对于给定数据源的关键词进行词频统计，并计算该关键词词频占比和给定股票价格之间的相关性。本文随机抽取了 10 个关键词，计算每个关键词 sigWordSeq 的累计运行时间，测试结果如图 4-2 所示：

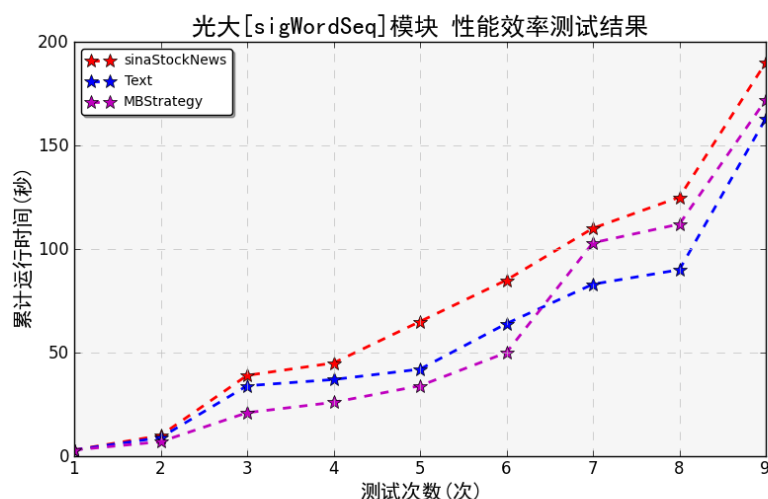


图 4-2 关键词词频模块 sigWordSeq 性能效率测试结果图

从图中可以看出，数据源个股新闻的平均耗时最高，平均每次运行时间约为 25 秒左右，数据源研究报告的平均耗时次之，平均每次运行时间约为 20 秒左右，数据源股票论坛的平均耗时是最低的，平均每次运行时间约为 18 秒左右。

3) 关键词网络模块 WordNet_stock

关键词词频模块中 WordNet_stock 的主要功能在于对于给定的关键词计算得到关联度最高的 20 只股票组合。本文随机抽取了 10 个关键词，计算每个关键词 WordNet_stock 的累计运行时间，测试结果如图 4-3 所示：

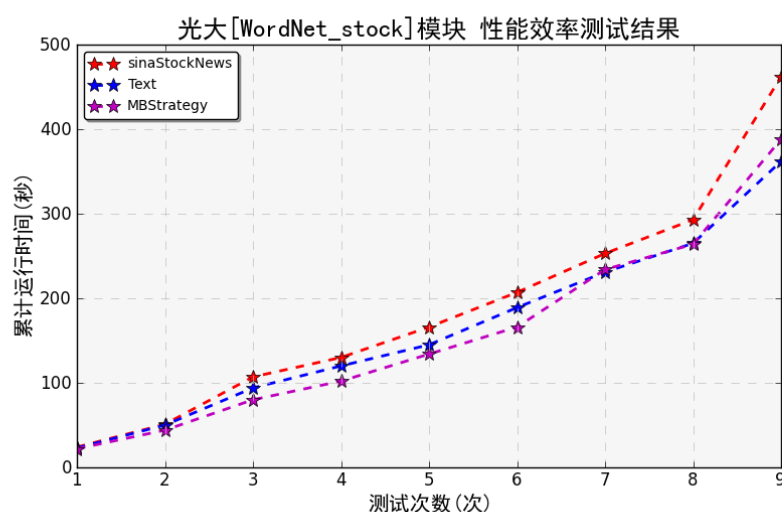


图 4-3 关键词网络模块 WordNet_stock 性能效率测试结果图

从图中可以看出，数据源个股新闻的平均耗时最高，平均每次运行时间约为 52 秒左右，数据源研究报告的平均耗时次之，平均每次运行时间约为 44 秒左右，数据源股票论坛的平均耗时是最低的，平均每次运行时间约为 40 秒左右。

4) 关键词网络模块 WordNet

关键词词频模块中 WordNet 的主要功能在于对于给定的关键词计算得到关联度最高的词语组成的网络图。本文随机抽取了 10 个关键词，计算每个关键词 WordNet 的累计运行时间，测试结果如图 4-4 所示：

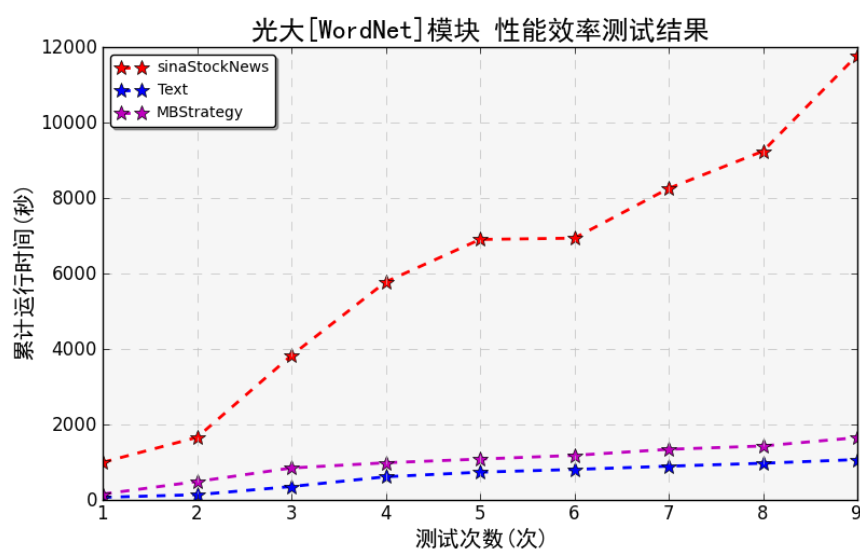


图 4-4 关键词网络模块 WordNet 性能效率测试结果图

从图中可以看出，数据源个股新闻的平均耗时最高，平均每次运行时间约为 20 分钟左右，数据源研究报告的平均耗时次之，平均每次运行时间约为 3 分钟左右，数据源股票论坛的平均耗时是最低的，平均每次运行时间约为 2 分钟左右。

5) 模块效率性能总汇

本文对于检索模块 SearchFiles，关键词词频模块 sigWordSeq、关键词网络模块 WordNet 和 WordNet_stock 的时间效率进行了实验。并以实验结果为依据，对每个模块三个数据源的平均运行时间进行了大致的评估，评估结果如表 4-1 所示：

表 4-1 智能文本分析系统各模块平均运行时间估计

	个股新闻	股票论坛	研究报告
检索模块 SearchFiles	25-30 秒	15-20 秒	20-25 秒
关键词词频模块 sigWordSeq	25-30 秒	15-20 秒	20-25 秒
关键词网络模块 WordNet_stock	50 秒左右	40 秒左右	45 秒左右
关键词网络模块 WordNet	20 分钟左右	2 分钟左右	3 分钟左右

根据上表所示的各模块平均时间效率估计的结果，可以得到如下结论：

- 在三个数据源中，所有模块个股新闻的平均运行时间是最长的，研究报告次之，而股票论坛是耗时最少的
- 所有模块的时间消耗主要都关键词的搜索上，模块的平均耗时和模块进行的关键词搜索次数成正比
- 关键词词频模块 sigWordSeq 进行了一次词频检索，因此和检索模块 SearchFiles 的平均耗时相当
- 关键词网络模块 WordNet_stock 同样只进行了一次关键词检索，但是在计算关联股票 TF-IDF 指标是需要耗费一定的时间，因此平均耗时略长于单次的检索
- 关键词网络模块 WordNet 由于需要进行对 20 个一级词的搜索，因此耗费的时间是最长的。另外，由于三个数据源中【个股新闻】的数据量最大，运行 WordNet 一旦遇到高频词会消耗大量的时间，需要格外注意

注意： 本文所进行的效率测试，只取了 10 个样本，实验所得的结果能定性的说明一些问题，实验对于时间效率也是做的测算也是基于该样本实验，可以作为对每个模块平均运行时间的大致估计，但每次系统各模块的具体运行时间可能会造成较大差异，这是由关键词在文本中出现的频数所决定的。