# 《数据科学与工程算法基础》实践报告

报告题目： 网络“水军”--基于爬山算法的抽取式摘要模型

姓 名：

学 号：

完成日期： 1.6

提供中英文摘要

摘要 [中文]：

本次项目爬取同花顺金融服务网的1000篇股票分析报告，应用tf-idf提炼关键词以及爬山算法生成抽取式文档摘要。项目旨在帮助快速生成宏观经济门户网站中的研报摘要同时也可以充当小型的网络水军。除了完成摘要的抽取工作，该项目还分析了方法的有效性和高效性以及研究中的不足和未来可能的研究方向，有利于更深入的研究的开展。

Abstract [英语]

This project crawls **1000** stock analysis reports of tonghuashun financial service network, extracts keywords by TF-IDF and generates extraction document summary by mountain climbing algorithm. The project aims to help quickly generate the research summary in the macro-economic portal website, and it can also serve as a small water army. In addition to the summary extraction work, the project also analyzes the effectiveness and efficiency of the method, as well as the shortcomings in the research and the possible future research direction, which is conducive to further in-depth research.

1. 项目概述（阐明该项目的科学和应用价值，以及相关工作进展并描述项目的主要内容）

本次项目爬取同花顺金融服务网。由于我对股票分析比较感兴趣，而浏览其中一篇一篇报告，缺乏关键词和摘要信息，很难快速了解我所关注内容的一些核心思想和观点，无法一目了然地明白这个文件的主旨。本项目拟开发一个智能应用助手，帮助人们快速生成宏观经济门户网站中的研报摘要，自动生成关键词和摘要。同时可以充当小型的网络水军，向百度等网站发送生成的摘要以提高个人博客网站的点击率和关注度。

1. 问题描述（问题定义）

**2.1符号定义**

|  |  |
| --- | --- |
| wpsoffice,i=1...n | 从门户网站收集1000个文档，每个文档用wpsoffice表示，所有文档组成文档集D |
| K | 所有文档生成的关键词集合 |
| wpsoffice,i=1...m | 一个wpsoffice表示一个句子，S表示所有句子组成的集合，m表示所有文档的句子个数 |
| wpsoffice,i不定 | wpsoffice表示摘要中的一个句子（该句子有包含关键词的原文档中抽取得到），R表示一个摘要 |

**2.2问题描述**

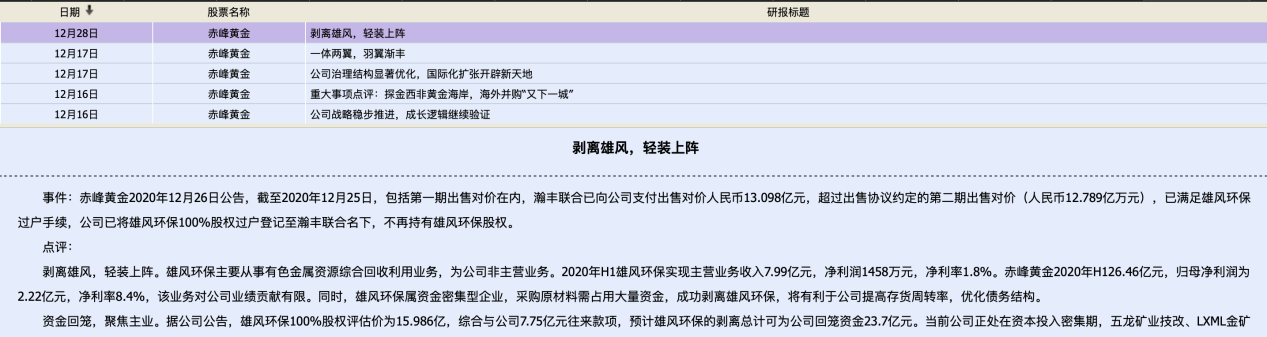
从同花顺金融服务网爬取1000个文档，利用TFIDF技术对文档集合D生成关键词集合K。将所有文档的句子汇总成集合S。从文档中选择尽可能少的句子ri尽可能多地覆盖关键词集合中的词语。每次选定生成摘要的集合随机，最后生成10篇关于宏观经济的摘要。

1. 方法（问题解决步骤和实现细节）

**3.1文档爬取**

本次项目选择爬取门户网站。爬取的同花顺金融服务网的结构和报告页面信息如下，可以看到比较简洁。每一页有10篇报告，总共爬取100页共1000篇文档。





由于网站有页码，获取每一个目录页。并打印url，可以看到url格式统一且固定。

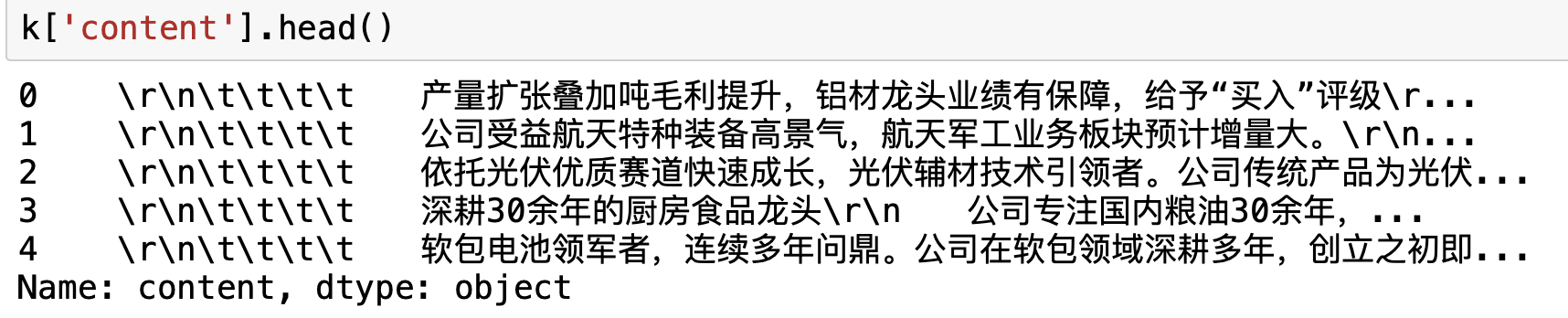


对于每一个url通过get\_list函数获取其中每一个报告页面的url信息。再通过page\_info函数利用每个报告页的url爬取其上的文字信息，并汇总到csv文件中。get\_list和page\_info函数代码如下。



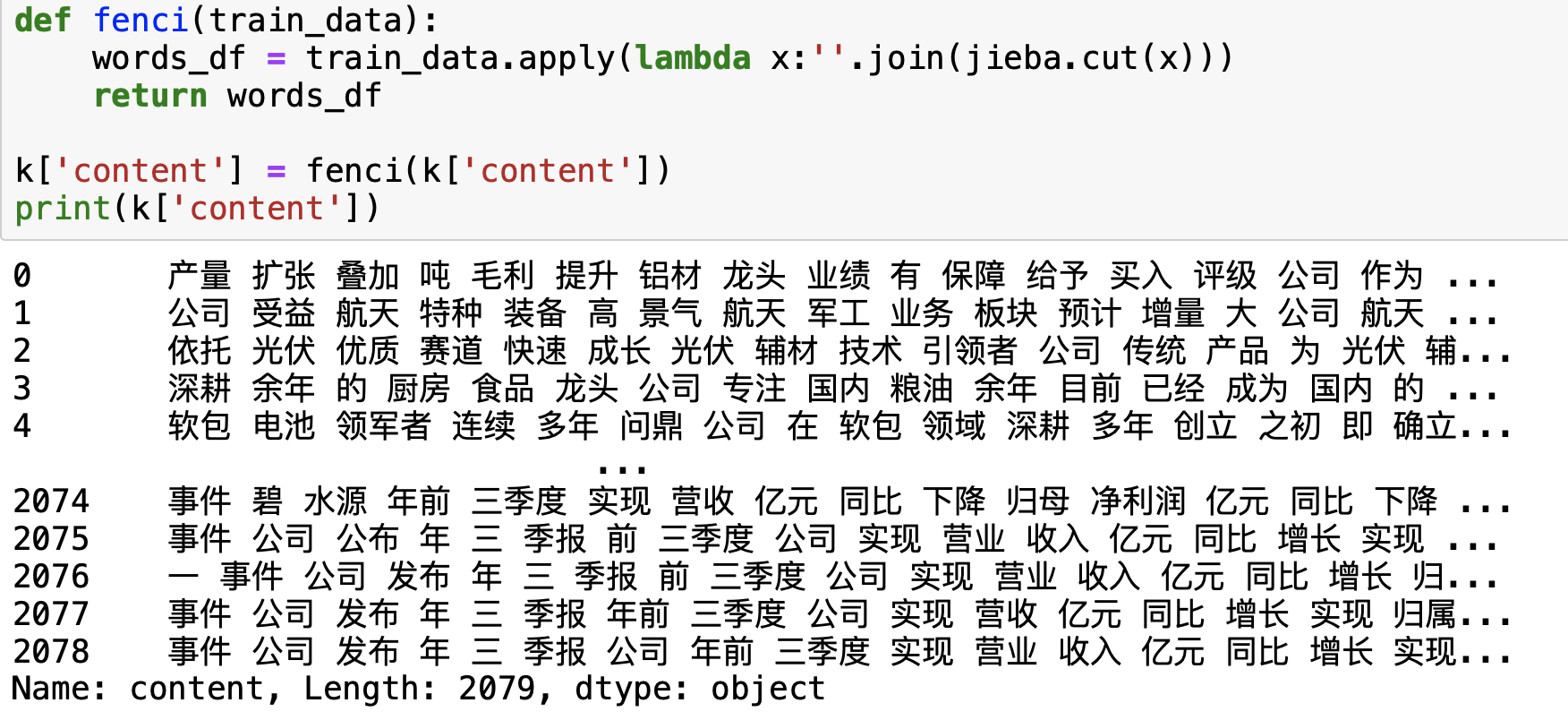
**3.2 数据预处理**

通过pandas的read\_csv读入需要处理的文档。打印后发现有很多多余的符号如” \r\n\t\t”，如下图所示：

通过正则表达式匹配将这些符号替换为空格。

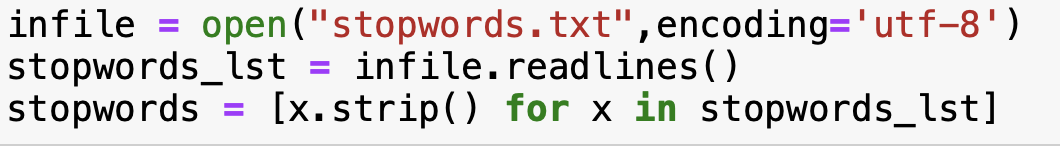
k['content'] = k['content'].str.replace(r'[^\u4e00-\u9fa5]','')

利用jieba库分词，由于jieba库的TFIDF可以直接去除停用词，故这里不进行停用词去除操作。分词代码和结果如下：

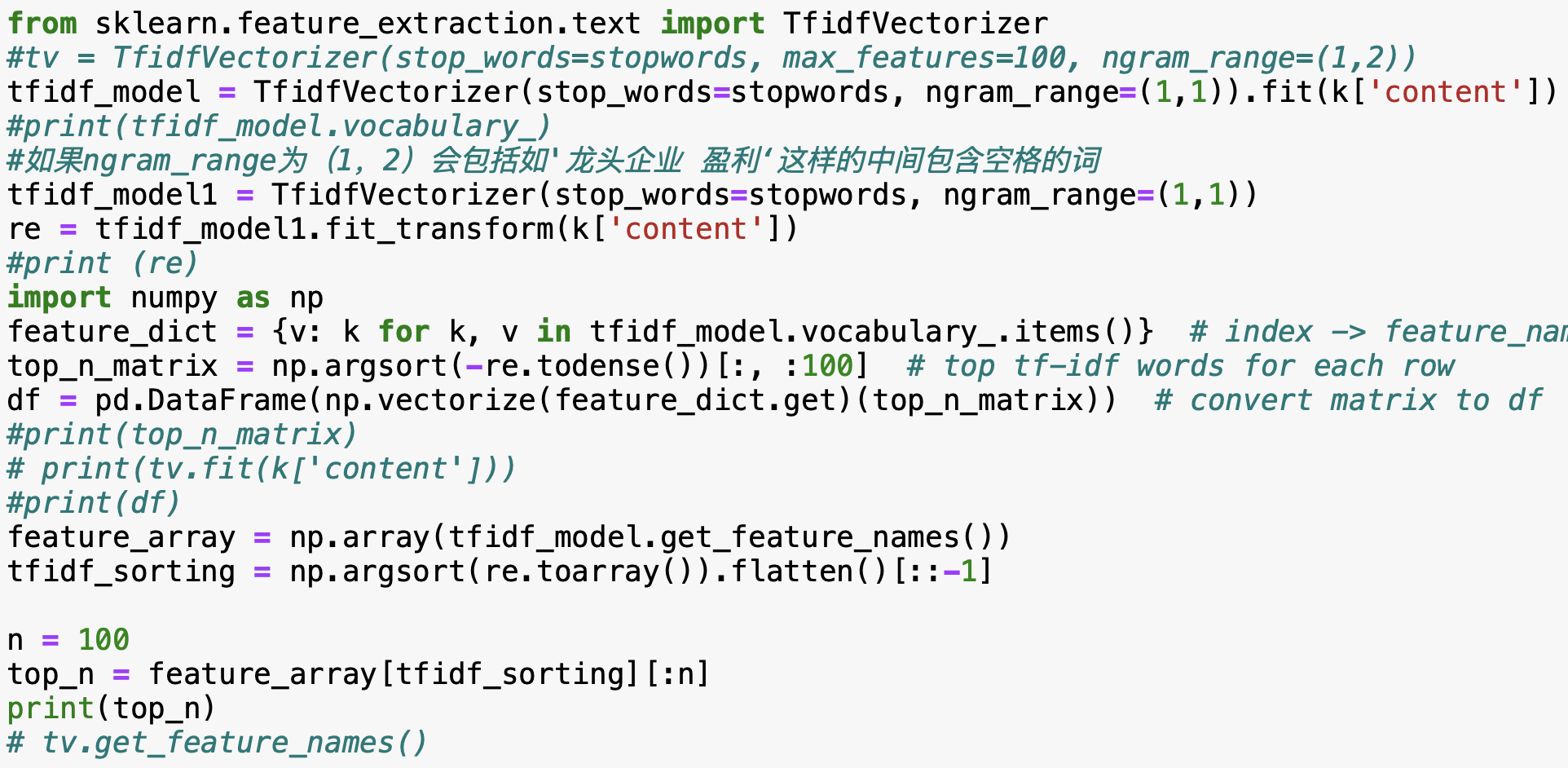


**3.3 关键词提取**

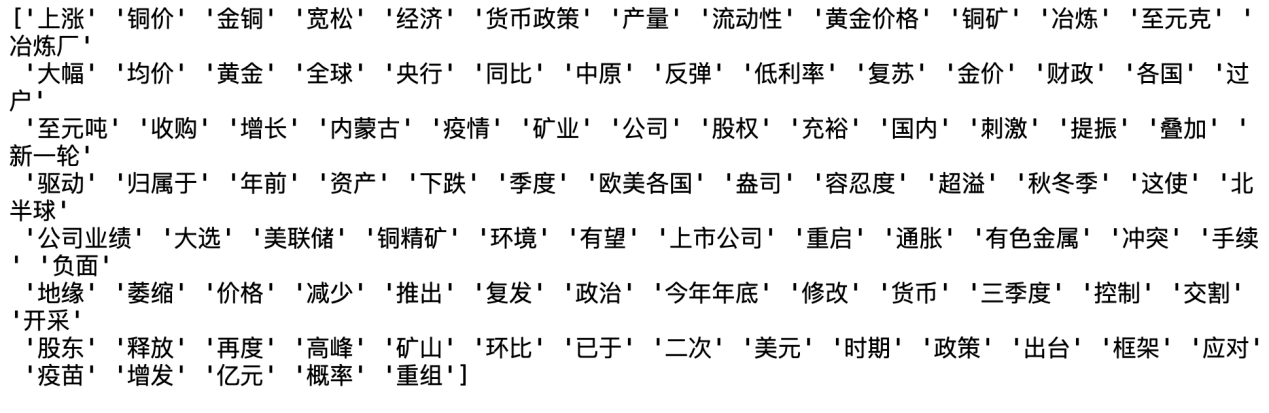
首先，先读入停用词，并保存在stop\_word中。



利用tfidfvectorizer进行TFIDF处理。参数中stop\_words为停用词词表。ngram\_range选为(1,1)仅保留单个词语。如果如果ngram\_range为（1，2）最后关键词结果中会包括如'龙头企业 盈利’这样的中间包含空格的词。通过调用tfidf\_model的get\_feature\_names得到关键词列表，排序，并选择排在前100的关键词作为最后的关键词集合。代码如下：

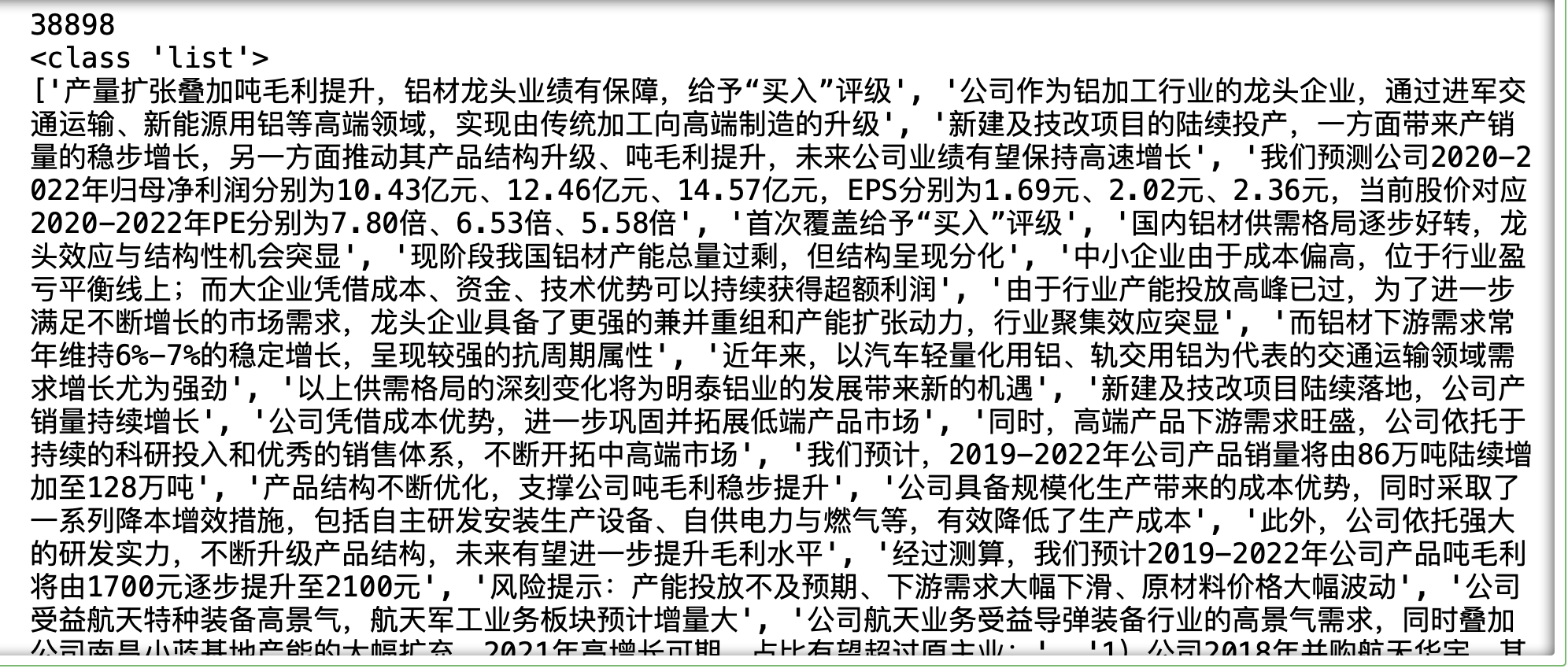


最后，得到前100个关键词如下：



**3.4 文档分句**

抽取式文档摘要是将包含关键词的句子抽取出来，为了方便抽取句子，提前做准备工作，将文档分句，并存入sentence列表中。打印分句结果如下图所示：



**3.5 爬山算法生成摘要**

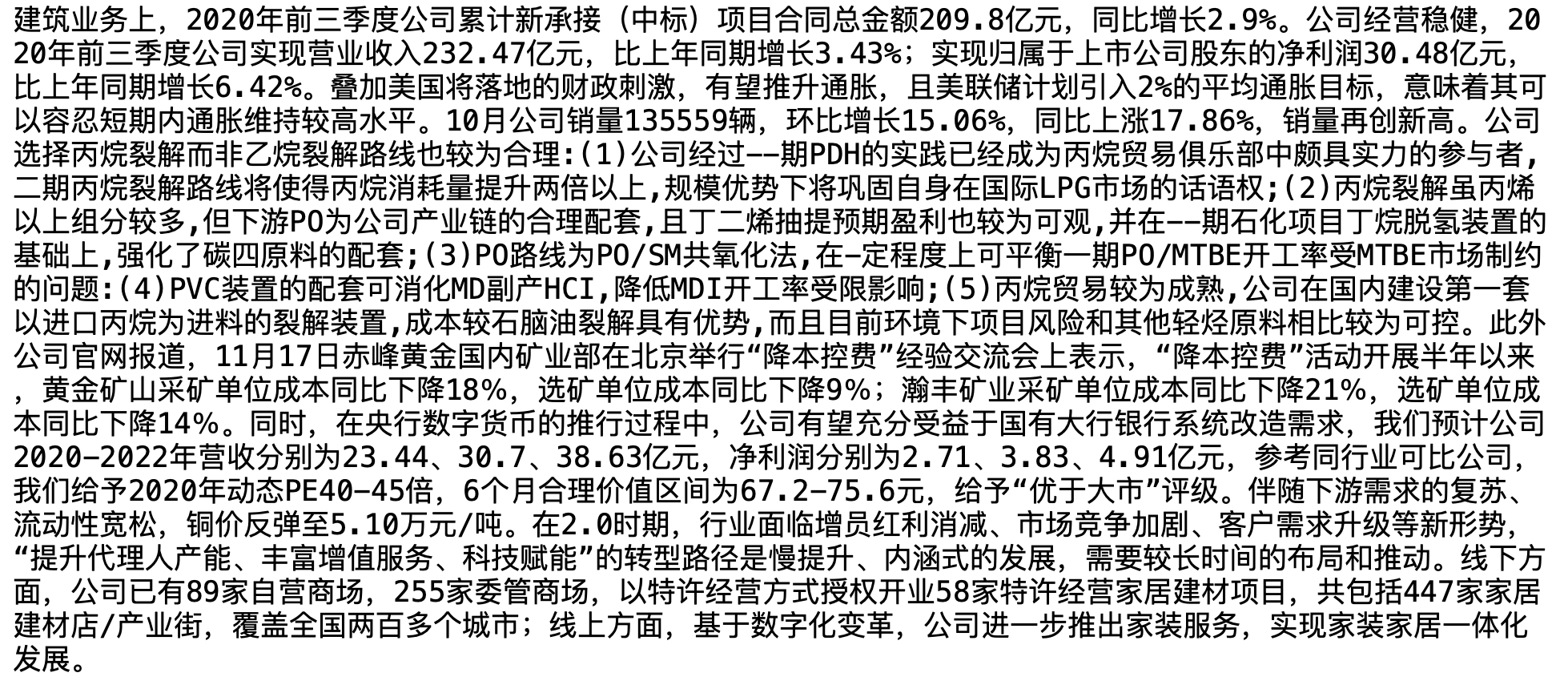
need\_cover为关键词集合。用selected标记被选择出的句子。summary为由所有selected的句子构成。选出10个句子保证这10个句子最大覆盖关键词集合。

首先通过分词生成每个句子包含的词语集合，计算其中包含的关键词，选出覆盖关键词最多的句子并将该句加入最后的summary中。

挑选出句子后，将其所包含的关键词从need\_cover中删除，进入下一轮迭代计算所有句子中包含更新后的need\_cover的关键词的数目，挑选的第二个句子。如果有多个句子包含关键词数目相同，则从中随机选择一个。

最后从初始状态共迭代10次得到充当网络水军的摘要。

生成的摘要如下：

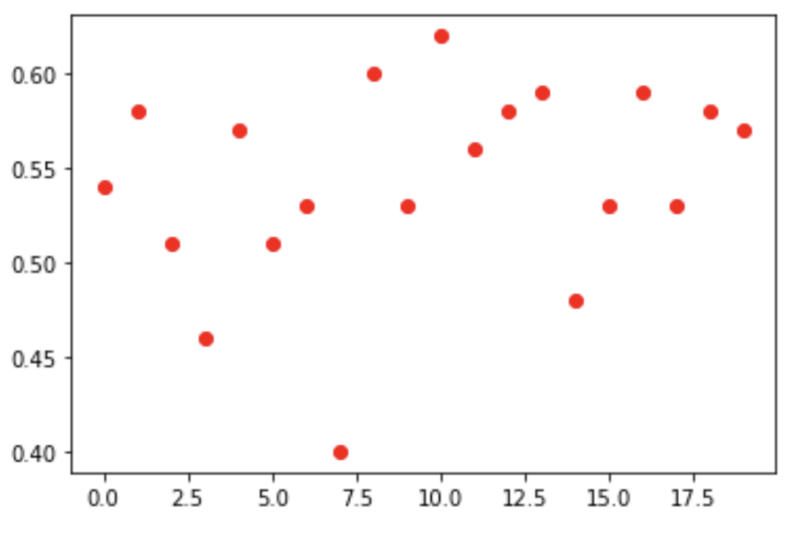


1. 实验结果（验证提出方法的有效性和高效性）

**有效性**

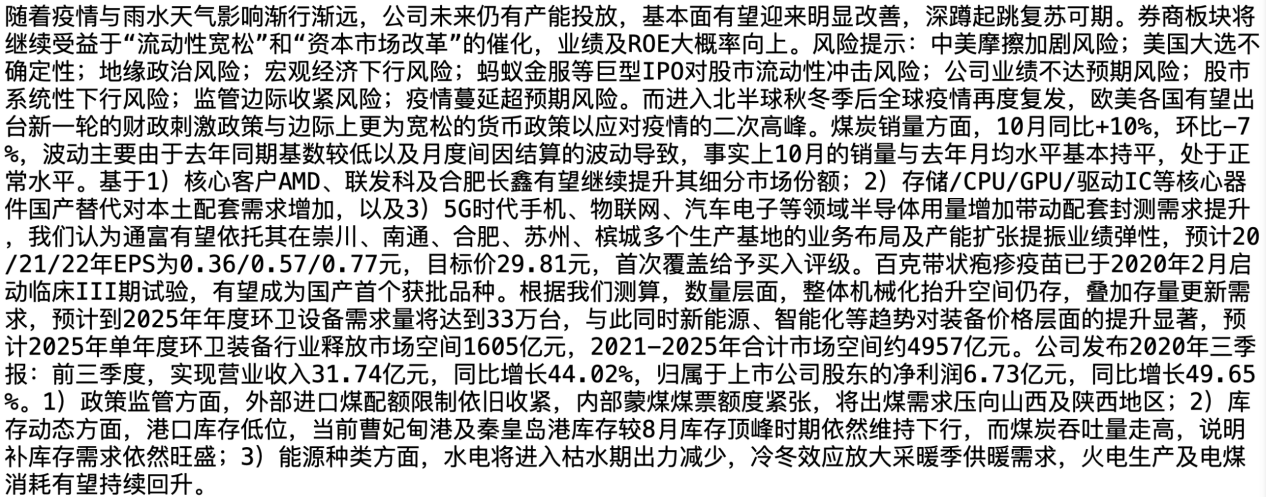
本次项目从生成摘要覆盖tf-idf计算出的出现频率高的关键词比例和生成摘要的可读性两个角度分析提出方法的有效性。

**关键词的覆盖率**：



本次项目采用选择10个句子的抽取式摘要力图最大覆盖原文档的高频关键词。上图显示的是随机生成的20篇摘要对所有文档中关键词的覆盖率，可以看到覆盖率集中分布在0.55-0.60之间，也就是通常情况下我们选择的10个句子中至少包含所有文档top100个关键词的55个左右。

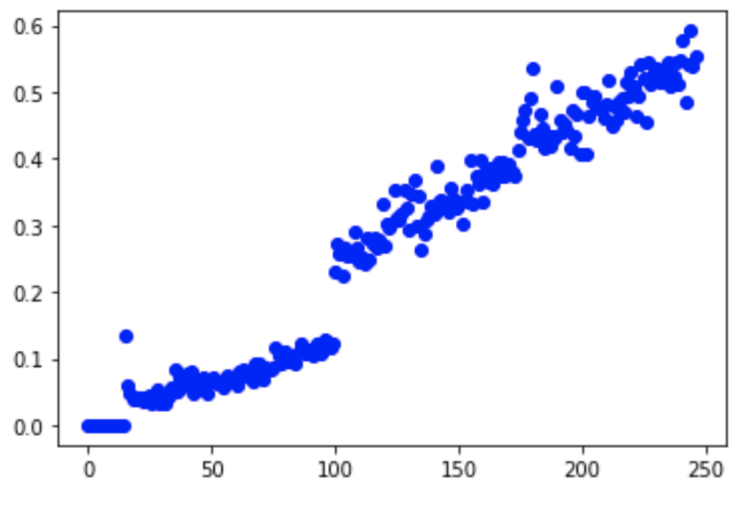
**摘要的可读性**：



随机从刚刚生成的20篇摘要选出一篇。可以看到其中有小标1）2）3）是有序排列的。标号之间的句子有;分割，最后一个句子末尾为句号。具有一定的可读性。如文中风险提示后包含了原文中的所有风险，风险之间用;分割。具有一定的连贯性。

**高效性**：

测算随着句子数的增加，程序生成10句话作为摘要的耗时。通过调用python中的time库，在程序运行前后分别记录start和end来求得最后的用时。其中测试时我们分情况讨论，如果句子数不到100，由于句子数较少，选择抽取5个句子生成摘要计算用时。句子数超过100时，抽取10个句子生成摘要计算用时。运行结果如下：



如上图所示，如上图比较了生成摘要的句子数对性能的影响可以看到句子数在100到150之间时呈平稳上升，上升幅度要高于50-100的上升幅度，在200-250之间有一个陡然上升的趋势。此外，可以看到生成摘要的用时总体都较短,该方法具有一定的高效性。

1. 结论（对使用的方法可能存在的不足进行分析，以及未来可能的研究方向进行讨论）

**方法可能存在的不足**：

1. 本文算法没有标准的摘要数据来计算BLEU和ROUGE这样更复杂的指标来度量最后的摘要对原始网页内容的覆盖程度。
2. 由于本文是抽取式模型，生成的摘要是已有的句子组成的，无法生成新的句子。

**未来可能的研究方向**：

1）基于NLG技术，根据源文档内容，由算法模型自己生成新的句子描述，而非提取原文的句子。将这种生成是模型的摘要与抽取式模型的摘要比较分析优劣

2）引入领域知识图谱优化关键词抽取的质量

3）采用现成的具有标准摘要的数据集计算各个复杂指标值

4）可以尝试其他的抽取式模型如基于图挖掘的各种Rank类算法与爬山算法的性能和结果质量进行比较