****新闻热点抽取系统****

****设计文档****

开发人员：

Houzss1

开发时间：2020/3/1 – 2020/6/1

新闻热点抽取系统 1

设计文档 1

第一章 绪论 6

1.1 研究背景 6

1.2 国内外研究现状 7

1.2.1文本向量化研究现状 7

1.2.2新闻文本聚类算法研究现状 8

1.2.3话题抽取算法研究现状 8

1.3 报告的研究内容 9

1.3.1基于K均值的层次凝聚方法的新闻文本聚类算法的设计 9

1.3.2热点话题抽取算法的设计 9

1.3.3新闻热点发现系统的实现 9

1.4 报告组织结构 10

第二章 新闻扒取模块 10

2.1 数据抽取 10

2.1.1基础知识 11

2.1.2所用技术/模型 11

2.1.3运行流程 13

2.2 文本预处理（新闻分词） 14

2.2.1基础知识 15

2.2.2所用技术/模型 15

2.2.3运行流程 16

第三章 新闻聚类模块 16

3.1基础知识 16

3.2所用技术/模型 17

3.2.1文本的表示 17

3.2.2 基于K-Means的层次凝聚算法 18

3.3运行流程 20

第四章 话题抽取模块 21

4.1基础知识 21

4.2所用技术/模型 21

4.3运行流程 21

第五章 新闻热点抽取系统的实现与分析 22

5.1 系统模块与流程 22

5.2 系统开发与实验环境 23

5.3 前后端设计与实现 24

5.3.1前端设计 24

5.3.2后端设计 25

5.4数据库设计与实现 26

5.5 文本抽取及存储模块实现 30

5.5.1 新闻抽取 30

5.5.2 新闻存储 31

5.6 文本聚类模块实现 34

5.6.1 文本向量实现 34

5.6.2 文本聚类模块的实现 35

5.7 热点话题抽取与热度值计算模块实现 38

5.8 功能性测试与分析 40

5.9 性能测试与分析（轮廓系数） 44

第六章 总结与展望 45

6.1 总结 45

6.2 系统未来展望 46

附录 参考文献 47

# 第一章 绪论

## 1.1 研究背景

随着互联网技术的蓬勃发展，人们获取新闻的方式逐渐从报纸、广播、电视等传统媒体转到新闻网站、新闻APP、微博微信自媒体等新兴互联网媒体。2018年６月，中国互联网信息中心发布了第42次《中国互联网络发展状况统计报告》，报告显示：至2018年6月30日，我国网民规模达到8.02亿，互联网普及率达57.5%。与此同时，移动设备也逐渐成为互联网的核心接入方式，移动互联网逐渐成为人们日常生活中必不可缺的部分。

从以上的数据可以发现，网络媒体作为一种新兴的媒体形式，已经逐渐成为人们日常获取新闻内容的主要形式，相比报纸、电视等传统新闻媒体，互联网涉及的内容更加广泛，更新速度更加迅速，有着无可比拟的极大优势，它可以快速、高效、广泛地传播新闻信息。随着移动互联网的快速发展以及类似于微博、微信等各种网络社交平台的普及，各种重大新闻事件的传播速度也将飞速发展，一件重大事件的产生到成为舆论的时间也将越来越短。

一般而言，一篇热点新闻从发生开始，在较短时间内就会被大量转载与更新，描述同一件事情的新闻会在网络中成百上千的出现，它们被分布于各类新闻网站中，为了有效的获取某一新闻的热度，需要设计高效的新闻热点话题发现方法，它应能够完成网络中的话题分类与热度值分析，提高网络资源的利用率。除此之外，随着互联网的普及，时间与空间作为舆论传播的限制被打破，舆论的演化和扩散将会瞬息万变，其新颖的内容在传播的同时很容易带有民众的主观意图和想法，如果热点舆论话题被不法分子煽动唆使，诱导不明真相的群众则会造成很大的社会影响和冲突。因此对于这些热点问题，新闻工作者和政府部门需要及早发现并跟踪，积极调查其新闻态势，了解民众的态度，分析总结新闻相关的舆情信息，以防其影响社会的和谐稳定，掌握舆情的正确基调。基于此，本文提出了一种基于k均值的层次凝聚聚类的新闻文本聚类方法，并基于该方法设计并实现了基于数据挖掘的新闻热点发现系统。

## 1.2 国内外研究现状

新闻热点发现系统中主要包含了文本向量化、新闻文本聚类算法、热点话题抽取算法三个模块，每个模块在文本挖掘的研究中都具有重要的意义，下面针对这三个部分分别进行概述。

### 1.2.1文本向量化研究现状

无论是在文本挖掘还是其他的自然语言模型中，文本的向量空间模型都是用来描述文本的常用方法，简单的采用分词算法和词频统计来表示文本将会导致文本的向量模型维度过大。高维数据不但会大大增加模型的计算开销而且冗余的数据也会影响模型的精度。因此采用文本向量化模型来处理文本是实现新闻热点挖掘系统的重要一步。

常用的文本向量化方法词典向量化模型以及ＮＮＬＭ的深度学习向量化模型。词典模型的向量化方法通过表征词典中的单词在文本中是否出现或出现的次数来向量化模型，其优点是简单可行，但是缺点是该方法生成的向量维度过大影响后续的文本处理。

为了改进词典模型向量维度过大的缺陷，学者们研究了基于降维的词典模型，将部分关键词提取来作为文本的代表。其中特征选择方法采用一种特殊的单词排序算法，并筛选出关键单词，采用关键单词向量来表示文本，单词特征排序可以采用文档频率(DF)、TF-IDF、互信息、卡方检验等方法来实现。基于TF-IDF方法，徐文海等人提出了一种基于向量空间模型的中文关键词提取算法，通过该方法实现了中文科技文献的关键词自动抽取。

近年来基于深度学习的向量化方法更为常用，Liu Ｙ等人于2015年提出了一种Topic Embedding方法，相对于word2vec等单词的向量化方法，Topic Embedding实现了句子乃至文本的直接向量化。Kim Y等人提出了一种可以直接基于Embedding的文本相似度比较方法，通过RNN-LM模型来实现单词的向量化，并在文本相似度比较上得到了显著的效果。

上述向量化方法在一般文本上都有着较好的效果，但是对于新闻文本而言，尤其是Topic Embedding这种直接生成文本向量的模型将会丢失大量的新闻细节，而且对于文本中的冗余信息也难以处理。

### 1.2.2新闻文本聚类算法研究现状

文本聚类是聚类任务的一个重要组成部分，近年来随着互联网发展，每天在互联网上海量生成新闻文本，基于对这些新闻文本进行数据挖掘的需求，新闻文本聚类的研究也逐渐火热。

常见的新闻文本聚类方法均采用了向量模型，李欣雨等人提出的向量模型聚类方法主要通过改进TF-IDF来实现聚类模型的优化，在聚类准确率上得到了提升，但是该方法的实际优化仅仅局限于文本相似度比较上的优化，在聚类模型上没有实现优化，优化程度有限。

基于密度聚类方法的新闻文本聚类方法也有许多学者进行研究。刘颖莹等人提出了基于粒子群算法的新闻文本聚类算法，孙红光等人提出了一种基于改进Single-Pass的新闻文本聚类算法，刘宏超等人提出了一种基于DBSCAN的密度聚类算法，这些算法均是以传统的密度聚类方法或改进的密度聚类方法实现文本聚类，但是相对而言，这些方法在新闻文本预处理上没有进行针对性工作，这类密度聚类算法在新闻文本的聚类性能上存在提升空间。

### 1.2.3话题抽取算法研究现状

常用的话题抽取算法包括基于TF-IDF的关键词抽取方法和基于LDA主题模型的主题生成。Wei X等人于2006年提出了一种基于LDA主题模型的信息检索方法，可以有效地从文本数据中获取关键信息。Ramage D等人提出了一种Labeled-LDA模型，使得主题和用户之间可以得到关联，并可以将主题与特定单词进行联系，在相关测试上，该方法的性能相对于传统的LDA模型而言得到了较大的提升。就TF-IDF方法而言，Chen S等人提出了基于TFIDF-KE算法的微博主题抽取方法，该方法是TF-IDF算法对ITEM优化的算法，在相关测试上，基于该方法可以大大提高聚类的性能。陈仕吉在文献簇主题识别中尝试了TF-IDF算法，并设计了一种基于C-Value和TF-IDF组合的方法，他们还改进了原始算法不合理的地方，实现了中文引文的主题抽取。上述算法的话题抽取中，TF-IDF依然局限于关键词抽取，LDA主题模型则对新闻文本特征没有加以利用，均有一定的改进空间。

## 1.3 报告的研究内容

本文对新闻热点发现系统相关技术进行了设计并实现了新闻热点发现系统。报告主要包括三个方面，第一个方面是是基于K均值的层次凝聚方法的新闻文本聚类算法的设计、第二个方面是热点话题抽取算法的设计、第三个方面是对新闻热点发现系统的实现。

### 1.3.1基于K均值的层次凝聚方法的新闻文本聚类算法的设计

通过计算TF-IDF的值来计算各文本的文本向量，利用余弦距离来计算各文本之间的相似度。利用K-Means聚类算法和层次凝聚算法实现基于K-Means的层次凝聚方法的新闻文本聚类算法。

### 1.3.2热点话题抽取算法的设计

热点话题抽取主要是对聚类后的话题列表进行关键词提取和热度值计算，是对于聚类后的结果进行一个具体的呈现。考虑到了词向量维数的庞大，采用复杂度高的例如LDA模型、将会耗费大量时间而结果并没有太大差别，决定主要采用了textrank的关键字提取算法和针对簇内文章的数量和时间进行的热度值计算。

### 1.3.3新闻热点发现系统的实现

本文的新闻热点发现系统包括新数据源获取模块，核心算法模块及用户界面模块，其中核心算法模块集成了新闻文本获取、新闻文本聚类、热点话题抽取三个关键步骤。用户界面模块实现了与算法功能对应的用户页面，在页面中可以简便地调用相关步骤并直观地展示新闻热点、关键词与之相应的排名和热度值。从相关测试结果表明，本系统满足了预期的功能需求。

## 1.4 报告组织结构

文章的主要内容如下：

第一章，阐述课题的研究背景和研究意义，并调研在新闻热点发现系统中使用的相关核心技术在国内外的发展现状，大致介绍了论文的研究内容、创新点及论文的组织结构。

第二章对热点发现系统的相关技术进行了简单的介绍，介绍了数据挖掘的相关概念和常用的模型与算法，介绍了文本预处理中的文本分词及文本向量化表示方法；在聚类技术的介绍中，分别介绍了相似度计算方法与常用的聚类模型。

第三章是新闻热点发现系统的关键算法模型，包括文本标签向量聚类模型及热点话题抽取模型两个部分。文本标签向量聚类模型中，针对新闻文本篇幅较长、标准化表示困难的问题，本文采用了k均值的层次凝聚聚类的文本聚类方法，减小算法复杂度的同时也优化了算法的性能。热点话题抽取模型部分基于TextRank模型设计了热点话题挖掘方法，并设计了基于文章的总数与时间的热度值计算公式，实现了新闻热点话题热度的直观展示。

第四章是新闻热点发现系统的实现与测试，该部分详细介绍了系统的整体结构，以及三个关键模块的实现方法与关键代码。并对新闻热点发现系统进行测试，测试部分包括功能性测试和性能测试两个部分，并对测试结果进行了分析。

第五章是总结与展望，梳理了文章的研究内容并提出了部分改进的意见。

最后是附录：参考文献与源代码。

# 第二章 新闻扒取模块

## 2.1 数据抽取

随着互联网的迅速发展，信息产业相关领域随即产生了处理海量数据的业务需求；数据挖掘也随着大数据技术的发展逐渐引起信息产业界的极大关注，其主要原因是存在的海量数据因为媒介、格式、内容等差异难以使用。数据挖掘试图采用科学方法高效、有效地将这些海量数据转换成有用的信息和知识。数据挖掘已经在各种信息和知识领域得到了广泛的应用。本文中设计的新闻热点发现系统是基于数据挖掘中的文本挖掘设计的，文本挖掘是数据挖局的一个重要分支，其旨在采用数据挖掘的方法从海量文本数据中提炼有价值的信息或知识。

### 2.1.1基础知识

随着网络的迅速发展，万维网成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。搜索引擎(Search Engine)，例如传统的通用搜索引擎AltaVista，Yahoo!和Google等，作为一个辅助人们检索信息的工具成为用户访问万维网的入口和指南。但是，这些通用性搜索引擎也存在着一定的局限性，如：

(1)不同领域、不同背景的用户往往具有不同的检索目的和需求，通过搜索引擎所返回的结果包含大量用户不关心的网页。

(2)通用搜索引擎的目标是尽可能大的网络覆盖率，有限的搜索引擎服务器资源与无限的网络数据资源之间的矛盾将进一步加深。

(3)万维网数据形式的丰富和网络技术的不断发展，图片、数据库、音频、视频多媒体等不同数据大量出现，通用搜索引擎往往对这些信息含量密集且具有一定结构的数据无能为力，不能很好地发现和获取。

(4)通用搜索引擎大多提供基于关键字的检索，难以支持根据语义信息提出的查询。

为了解决上述问题，定向抓取相关网页资源的聚焦爬虫应运而生。聚焦爬虫是一个自动下载网页的程序，它根据既定的抓取目标，有选择的访问万维网上的网页与相关的链接，获取所需要的信息。与通用爬虫(general purpose web crawler)不同，聚焦爬虫并不追求大的覆盖，而将目标定为抓取与某一特定主题内容相关的网页，为面向主题的用户查询准备数据资源。

聚焦爬虫主要解决三个问题：

(1) 对抓取目标的描述或定义；

(2) 对网页或数据的分析与过滤；

(3) 对URL的搜索策略。

### 2.1.2所用技术/模型

(1) python-requests库

requests库是Python语言编写，基于urllib，采用Apache2 Licensed开源协议的HTTP库。它相对于urllib更加方便，大大节约了代理量，完全满足了HTTP测试相关需求。

(2)python-BeautifulSoup库

HTML 文档本身是结构化的文本，有一定的规则，通过它的结构可以简化信息提取。于是，就有了lxml、pyquery、BeautifulSoup等网页信息提取库。一般我们会用这些库来提取网页信息。BeautifulSoup是一个高效的网页解析库，可以从 HTML 或 XML 文件中提取数据。Beautifulsoup支持不同的解析器，比如，对HTML解析，对XML解析，对HTML5解析。一般情况下，我们用的比较多的是 lxml 解析器。当前最新的 Beautiful Soup 版本为4.4.0，Beautiful Soup 3 当前已停止维护。

(3)python-re库

正则表达式是一个特殊的字符序列，它能帮助你方便的检查一个字符串是否与某种模式匹配。Python 自1.5版本起增加了re 模块，它提供 Perl 风格的正则表达式模式。re 模块使 Python 语言拥有全部的正则表达式功能。compile 函数根据一个模式字符串和可选的标志参数生成一个正则表达式对象。该对象拥有一系列方法用于正则表达式匹配和替换。re 模块也提供了与这些方法功能完全一致的函数，这些函数使用一个模式字符串做为它们的第一个参数。

(4)python-json库

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，易于人阅读和编写。

### 2.1.3运行流程

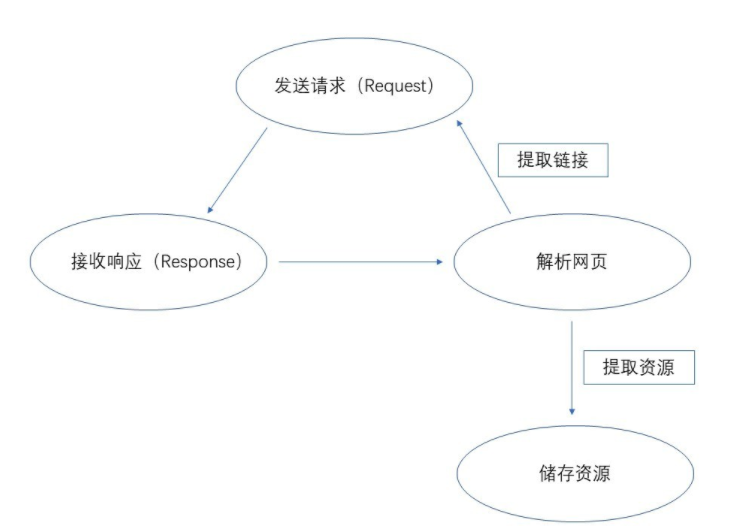


图2.1 requests获取网页资源流程图

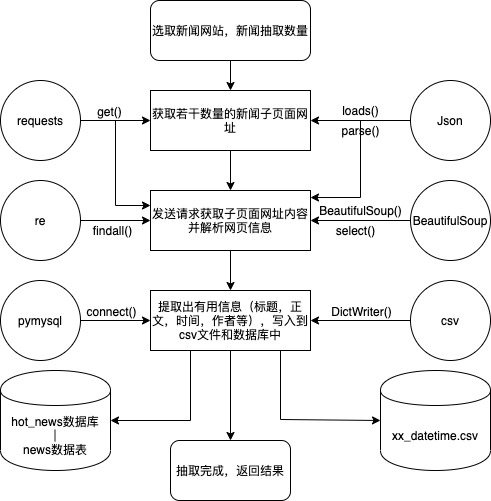


图2.2 新闻抽取流程图

## 2.2 文本预处理（新闻分词）

在文本挖掘任务中，如何有效的处理文本数据，使文本数据在计算机系统及

挖掘算法中得到更加直观的表现和更为优秀的性能是一个值得关注的问题;无论

是何种数据信息，在数据存储、数据运算中都需要按照合适的、标准的方法进行

表示;基于上述需求，文本预处理及相关技术是文本挖掘中的重要一部分，本文

中的新闻热点抽取系统采用的文本预处理技术主要包含了文本分词及文本去停用词两个步骤。

### 2.2.1基础知识

中文分词(Chinese Word Segmentation) 指的是将一个汉字序列切分成一个一个单独的词。分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。最常见的分词算法可以分为三大类：基于字符串匹配的分词方法、基于理解的分词方法、基于统计的分词方法。

停用词按照词性区分，主要分为四大类:一是连词；二是叹词；三是介词；四是标点符号。当然还有其他的一些词性的词属于停用词的范畴。本文采用的停用词表结合百度停用词表、四川大学停用词表、中科院停用词表等多个版本，综合考虑进行合并删除等处理后得到，能够涵盖大部分停用词。

### 2.2.2所用技术/模型

本项目采用python开源中文分词组件—jieba分词。对文本进行分词以后，发现很多文章中的词对于篇章语义表达没有太多的意义，对于文档和用户需求价值不大，需要将这些词项从词汇表中去除，比如连词、叹词、介词、标点符号等。这些词成为停用词( stopword)，在新闻中占了一定的比例，对于降低文档的维度有一定的帮助，下图2.3给出了一部分停用词。

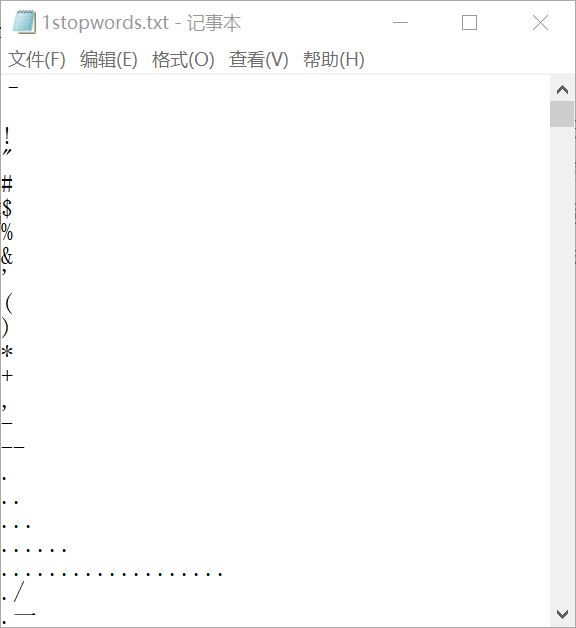


图2.3 停用词表

### 2.2.3运行流程

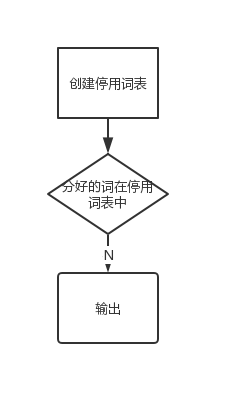


图2.4 去停用词流程图

# 第三章 新闻聚类模块

## 3.1基础知识

凝聚层次聚类法和K-Means算法是目前应用较为广泛的文本聚类方法。凝聚层次聚类法可能达到较高的精度，但时间复杂度较高，而K-Means则与之相反，具有较快的聚类速度，但是精度却较低。为此结合凝聚层次法和K-Means的特点，提出了一个新的基于K-Means的文本层次聚类算法。实验结果表明，该方法优于凝聚层次算法方法。且它的时间复杂度介于凝聚层次法和K-Means之间，有较好的实用性。

## 3.2所用技术/模型

### 3.2.1文本的表示

1).计算词向量

将爬取到的新闻文本内容（即数据库中列）进行分词去停用词之后形成一个词袋（用字典表示） ，则每个新闻文本都可以生成一个维的向量来表示。的计算方法为：设为在当前新闻文本中出现的次数，为总新闻文本中的问句个数，为新闻文本的总数，那么

a).是词频

词频（）表示词条（关键字）在文本中出现的频率。

公式：

即*：*

其中是该词在文件中出现的次数，分母则是文件中所有词汇出现的次数总和；

b).是逆向文件频率

逆向文件频率()：某一特定词语的，可以由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。

如果包含词条的文档越少, 越大，则说明词条具有很好的类别区分能力。

公式：

分母之所以+1是为了避免分母为零的情况

c).实际上是:

1. .计算文本相似度

新闻文本之间的相似度可以转换为向量之间的距离来进行度量。距离越小，新闻文本之间的相似性越大，反之亦然。

项目采用余弦夹角来计算向量之间的相似度，相似主题的新闻文本一般包含更多相同的特征词，两个新闻文本的主题是否接近，取决于它们的特征向量“长得像不像”。两个新闻文本的相似度可以利用这两个向量之间夹角的余弦值来表示。

相似度的计算公式如下所示：

其中表示两个向量的点积，表示向量的长度。

### 3.2.2 基于K-Means的层次凝聚算法

1. .K-Means算法的一般描述

K-means算法是最普及的聚类算法，也是一个比较简单的聚类算法，所以刚接触的同学不要感到害怕。算法接受一个未标记的数据集，然后将数据聚类成不同的组，同时，k-means算法也是一种无监督学习。

k-means算法的思想比较简单，假设我们要把数据分成K个类，大概可以分为以下几个步骤：

1.随机选取k个点，作为聚类中心；

2.计算每个点分别到k个聚类中心的聚类，然后将该点分到最近的聚类中心，这样就行成了k个簇；

3.再重新计算每个簇的质心（均值）；

4.重复以上2～4步，直到质心的位置不再发生变化或者达到设定的迭代次数。

由于采用迭代方法，可能只能得到局部的最优解，而无法得到全局的最优解。所以代码设定中默认进行十次运行，取轮廓系数最大的局部最优解。并且K值需要预先设定，代码中对样本数量不同进行了区别设定。

2).层次凝聚算法的一般描述

凝聚层次聚类与K-Means所采取的划分策略不同，凝聚层次法是一种自底向上的方法。它首先将每个文档作为一个类(原子类)，然后合并这些类成为较大的类，直到所有的文档都在一个类别中，或者用户指定的某个终止条件被满足。不同层次凝聚法的主要区别在于类间相似度的定义方法不同。

一般来说，有三种常用的方法，即：single-link、complete-link和UPGMA(也被称为组平均Group Average)。对于两个类，它们之间的相似度分别定义为：

在本实验中采用组平均的层次凝聚算法，它的时间复杂度为。该算法调用sklearn库中的AgglomerativeClustering()模型，对于K值同样需要预先赋值。

1. .基于K-Means的层次凝聚算法

对于K-Means这样的划分的聚类算法，它的优点是当使用算法进行聚类的时候，可以利用整个文本集的全局信息。但是凝聚层次法要聚类的时候，则更多的利用的是文本集的局部特征。

使用文本集的局部特征既有优点也有缺点。优点是人们可以很方便地将文本聚合成一个较小且足够内聚的类；而K-Means之类的划分算法要想完成这项任务，则可能因为在划分聚类的过程中算法越过类的界限使这些文本分散而失败。缺点是：如果一旦某-步做出了错误的合并决定，由于凝聚层次法每步所做的处理不能被撤消，类与类之间也不能交换对象，因而这些错误会在以后的凝聚过程叠加，会导致低质量的聚类结果。这样的情形在有大量相当的替代选择时候尤为明显。

基于以上考虑，提出了一种基于K-Means的层次凝聚方法的新闻文本聚类算法，即：使用K-Means方法所产生的类来约束凝聚层次法的凝聚空间。

## 3.3运行流程

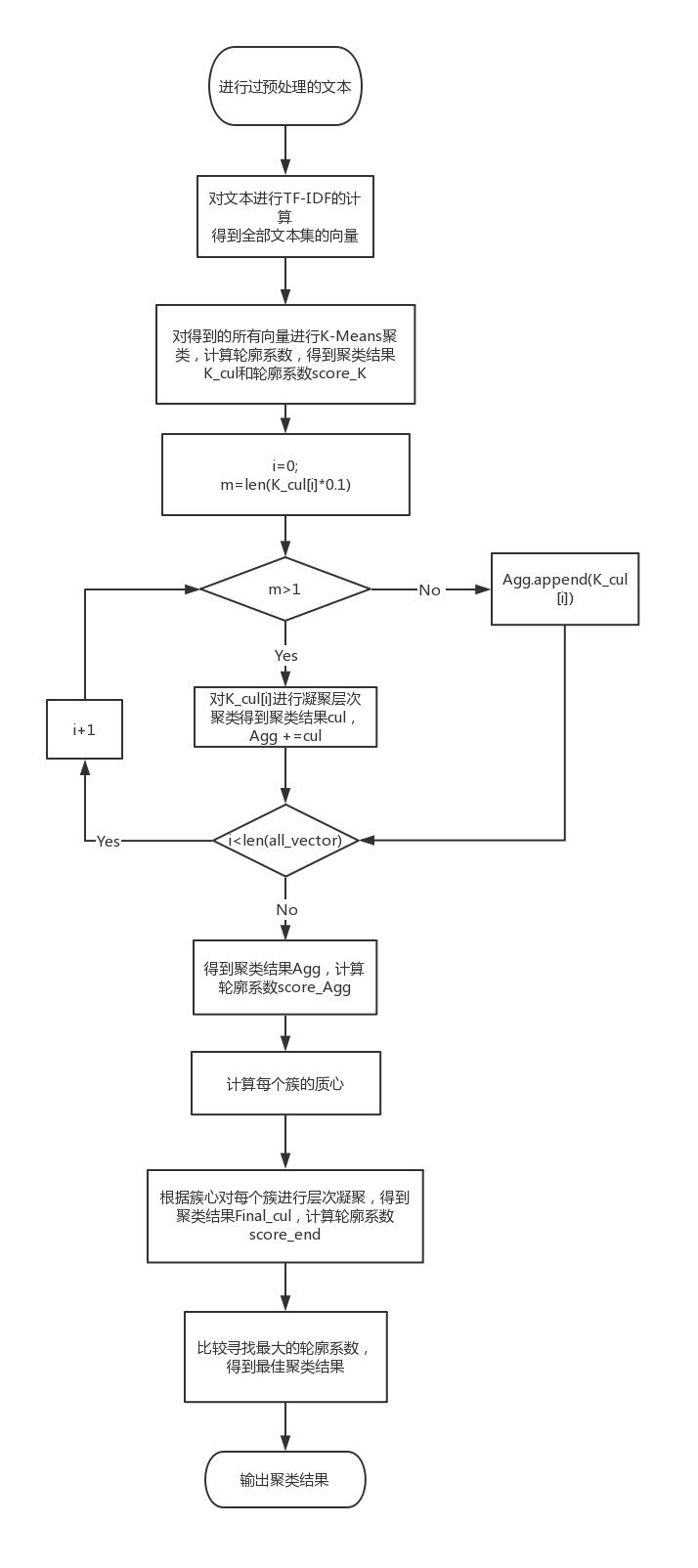


图3.1 新闻文本聚类流程图

# 第四章 话题抽取模块

## 4.1基础知识

Textrank算法核心是将原文本拆分为句子，在每个句子中过滤掉停用词，并只保留指定词性的单词。由此可以得到句子的集合和单词的集合。每个单词作为pagerank中的一个节点。设定窗口大小为k，假设一个句子依次由下面的单词组成：w1,w2,w3…wn。它们都是一个窗口。在一个窗口中的任两个单词对应的节点之间存在一个无向无权的边。基于上面构成图，可以计算出每个单词节点的重要性。最重要的若干单词可以作为关键词。

## 4.2所用技术/模型

textrank4zh库：

TextRank4ZH 用于自动从中文文本中提取关键词和摘要，基于 TextRank 算法，使用 Python 编写。TextRank 算法可以用来从文本中提取关键词和摘要（重要的句子）。TextRank4ZH是针对中文文本的TextRank算法的python算法实现。

## 4.3运行流程

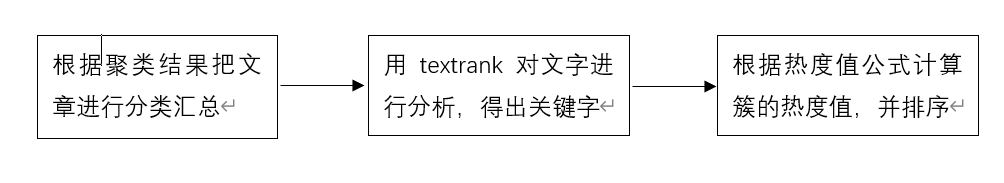


图4.1 话题抽取流程图

# 第五章 新闻热点抽取系统的实现与分析

## 5.1 系统模块与流程

新闻热点发现系统的目的是从海量新闻数据中挖掘出热点新闻话题，其整体架构分为文本数据获取模块、文本聚类模块、热点话题抽取模块。新闻热点发现系统的系统架构图如下图5.1所示。热点的发现过程如下：

１）文本数据获取模块从新闻门户网站中爬取新闻内容，经由预处理系统（jieba分词）处理完毕后保存于数据仓库中。

２）当发起话题抽取任务时，文本聚类模块按照指定时间从数据仓库中获取新闻文本，通过K均值的层次凝聚聚类算法为指定新闻文本打上对应的簇签。

３）热点话题抽取模块遍历每一个簇，采用Textrank算法获取对应簇的主题，并按照簇为单位存储对应的主题和该簇下的文本数目。

４）热度值计算模块对每一个主题计算热度值并保存。

５）前端展示界面调用指定批次抽取的新闻提取特定数量的关键词（热点话题）和热度值数据加以展示。

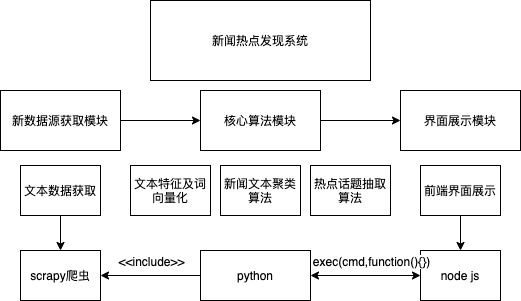


图5.1 系统模块流程图

## 5.2 系统开发与实验环境

服务器配置要求如表5.1所示。

|  |  |
| --- | --- |
| 配置项 | 配置参数 |
| CPU型号 | 2.3 GHz 双核Intel Core i5 |
| CPU主频 | 2.30 GHz |
| 内存容量 | 8GB |
| 硬盘容量 | 1TB |
| 显卡芯片 | Nvidia GeForce GT 620(1GB) |
| 操作系统 | Mac os Catalina 10.15.1 |
| 无线网卡 | 瑞昱 RTL8168/8111/8112 Gigabit Ethernet |

表5.1 服务器配置要求表

系统运行软件环境如表5.2所示。

|  |  |
| --- | --- |
| 环境项 | 环境参数 |
| 操作系统 | 具有Java虚拟机的操作系统均可 |
| 数据库 | MySQL |
| 浏览器 | 兼容Chrome/Firefox/IE等 |

表5.2 系统运行软件环境表

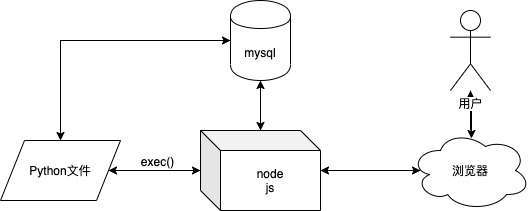


图5.2 系统环境架构图

文本抽取模块由Python 3.7开发，其中爬虫模块采用基于Python3的requests+BeautifulSoup爬虫框架，数据预处理模块采用了基于Python3的Numpy及Pandas的数学分析框架，分词模块采用基于Python的jieba分词工具,存储模块采用基于Python3的Pymsql存储于mysql数据库中。

文本聚类模块基于Python3.7开发，聚类模块调用了Python3下的sklearn,math,Bio等库开发完成。

话题抽取模块由Python3.7开发完成，其中Textrank基于调用Python3的Textrank4zh库函数。

测试环境为mac平台的Chrome浏览器，测试数据包括基于爬虫获取的人民日报293条数据。

## 5.3 前后端设计与实现

### 5.3.1前端设计

模块库：Bootsrap,Jquery,html,css,JavaScript等

大致页面及功能：

1.主页面

三大模块选择：新闻抽取任务，新闻聚类/话题抽取任务，历史任务查询；

2.新闻抽取：参数:新闻网站选取和抽取新闻数量。

3.新闻聚类： 选择相应抽取的新闻批次进行聚类分析（已完成聚类的批次不能重复进行聚类任务，只可以在历史记录中查询）

4.话题抽取： 从对应批次新闻完成的聚类中提取若干话题（关键词）并计算热度值，按大小倒序排名展现。

5.历史记录查询：查询历史进行的新闻抽取/聚类/话题抽取任务。



图5.3 主页功能图

### 5.3.2后端设计

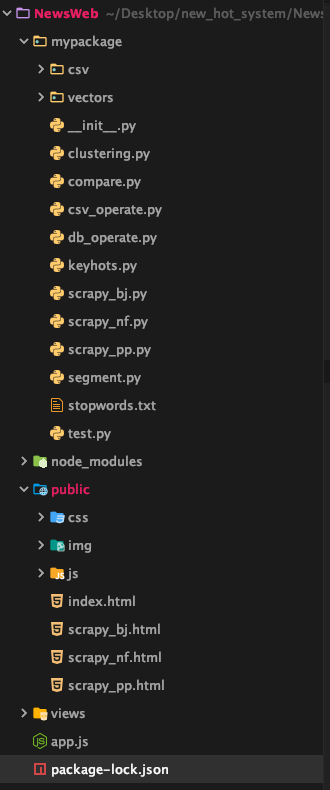


图5.4 项目代码结构图

框架：NodeJS

模块：Express,body-parser,cookie,ejs,json,mysql,parseurl等

过程：

1.通过路由完成对请求的处理、反馈与交互（/app.js）,包括执行python代码，与数据库交换数据，向请求数据的网页返回查询结果等。

2.通过JS完成前后端数据传输的中间处理（/public/js/diy.js），包括将后台返回的数据渲染生成网页页面展示。

## 5.4 数据库设计与实现

**数据库hot\_news:**

CREATE DATABASE Hot\_News DEFAULT CHARACTER SET utf8；

**1.存取新闻数据news:**

CREATE TABLE IF NOT EXISTS news (id varchar(50) PRIMARY KEY, title varchar(50) not null, txt LONGTEXT not null, datetime varchar(30) not null, source varchar(50) not null, web varchar(5) not null, batch varchar(50) not null)DEFAULT CHARSET=utf8;

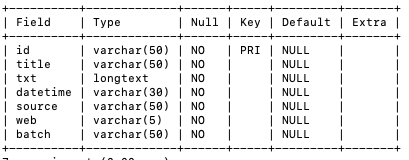


图5.5 news数据表

Id 序列号（batch + id :str)

Title 新闻标题

Txt 新闻内容

Datetime 新闻发布时间

Source 新闻作者/出版社

Web 来源网页

batch 批次号

**2.存取历史扒取任务his\_tasks(视图)：**

create view his\_tasks as select batch,concat\_ws('-',substring(batch,1,4),substring(batch,5,2),substring(batch,7,2)) as datetime,count(\*) as amount, web from news group by batch, web;

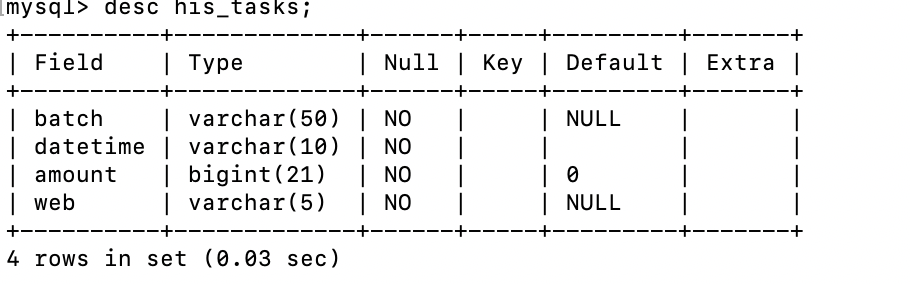


图5.6 his\_tasks数据表

批次号 batch (primary)

新闻数目 amount

扒取时间 datetime

来源网站 web

3.存取新闻聚类结果表clustering:

CREATE TABLE IF NOT EXISTS clustering(clusterid varchar(50) PRIMARY KEY, categoryid int(5) not null, clusternum int(11) not null, batch varchar(50) not null, vectors LONGTEXT not null, clusterbatch varchar(30) not null, web varchar(5) not null)DEFAULT CHARSET=utf8;

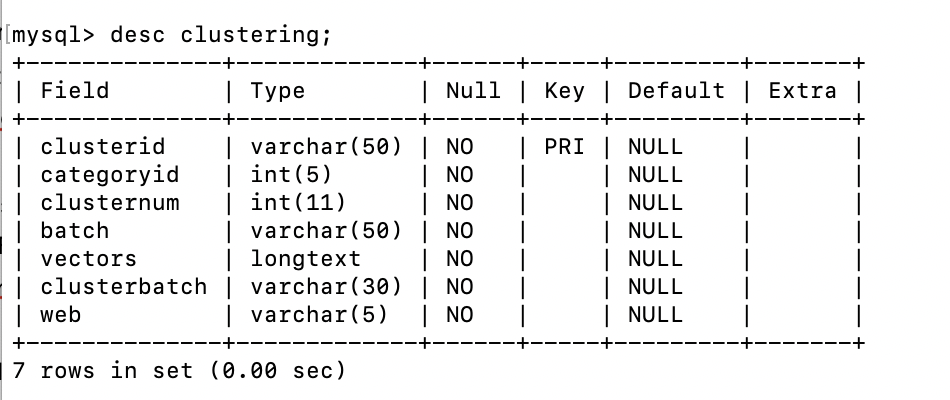


图5.7 clustering数据表

聚类ID（聚类时间+类序号）clusterid

类序号 categoryid

该类新闻数 clusternum

所属批次 batch

新闻向量 vectors

聚类批次 clusterbatch

来源网站 web

4.历史聚类任务结果his\_clusters(视图):

Create view his\_clusters as select batch as news\_scrapy\_batch ,clusterbatch,concat\_ws('-',substring(clusterbatch,1,4),substring(clusterbatch,5,2),substring(clusterbatch,7,2)) as datetime, web, sum(clusternum) as amount ,count(\*) as categorynum from clustering group by batch, clusterbatch,web;

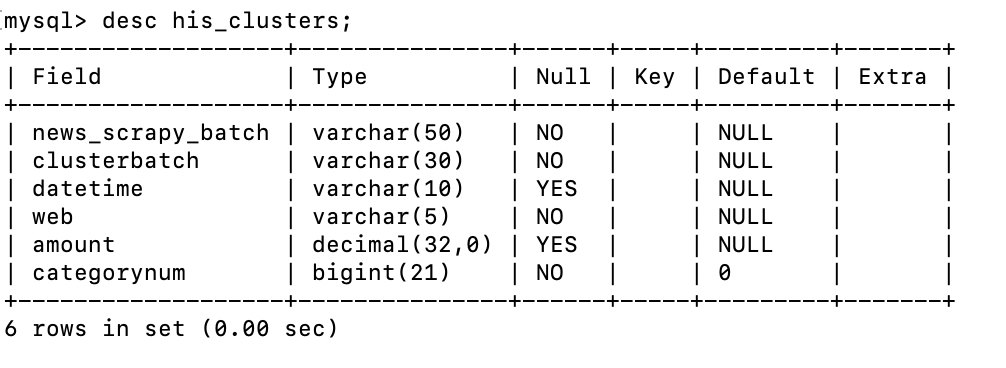


图5.8 his\_clusters数据表

新闻批次 news\_scrapy\_batch

聚类批次 clusterbatch

聚类日期 datetime

来源网站 web

新闻总数 amount

聚类数量 categorynum

5.剩余未聚类任务todo\_clusters(视图):

Create view todo\_clusters as select \* from his\_tasks where batch not in(select news\_scrapy\_batch from his\_clusters);

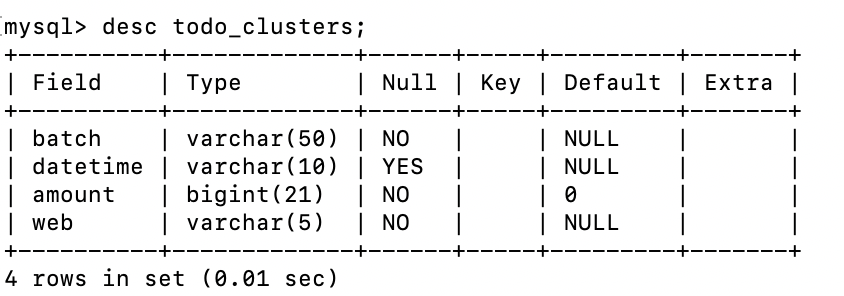


图5.9 todo\_clusters数据表

表项同his\_tasks;

6.存取新闻抽取完成后的关键字，热力值结果表keyhots

CREATE TABLE IF NOT EXISTS keyhots(keyhotsid varchar(50) PRIMARY KEY, categoryid int(5) not null, batch varchar(50) not null, keywords LONGTEXT not null, keywords\_num int(5) not null, hotvalues varchar(20) not null, keyhotsbatch varchar(30) not null, web varchar(5) not null)DEFAULT CHARSET=utf8;

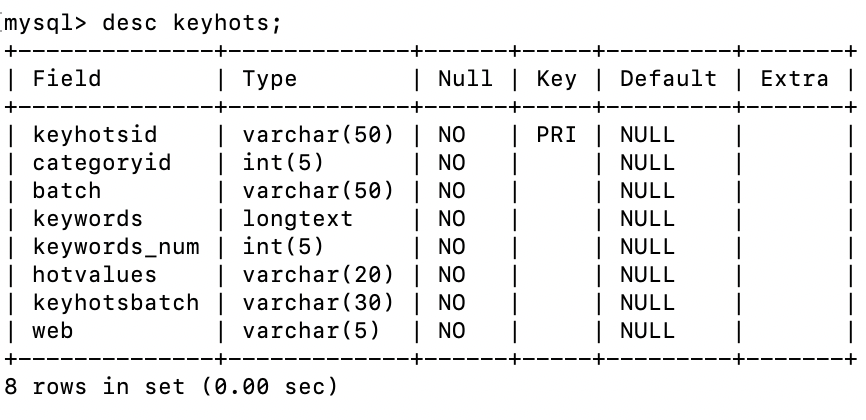


图5.10 keyhots数据表

抽取ID（抽取时间+类序号）keyhotsid

类序号 categoryid

所属批次 batch

关键字 keywords

关键词数 keywords\_num

热力值 hotvalues

话题抽取时间 keyhotsbatch

新闻来源网站 web

7.已完成新闻抽取历史记录 his\_keyhots(视图)

Create view his\_keyhots as select batch as news\_scrapy\_batch, keyhotsbatch, concat\_ws('-',substring(keyhotsbatch,1,4),substring(keyhotsbatch,5,2),substring(keyhotsbatch,7,2)) as datetime, web ,keywords\_num from keyhots group by batch, keyhotsbatch, web,keywords\_num;

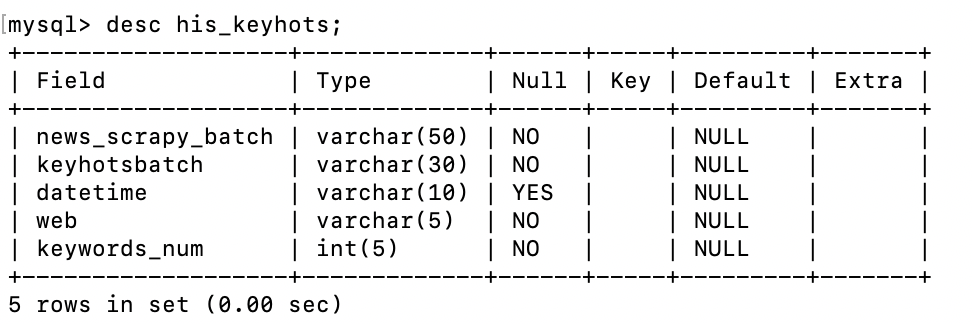


图5.11 his\_keyhots数据表

新闻批次 news\_scrapy\_batch

抽取批次 keyhotsbatch

抽取日期 datetime

来源网站 web

关键词数 keywords\_num

## 5.5 文本抽取及存储模块实现

### 5.5.1 新闻抽取

采用了Requests+BeautifulSoup来实现新闻抽取，核心代码为：

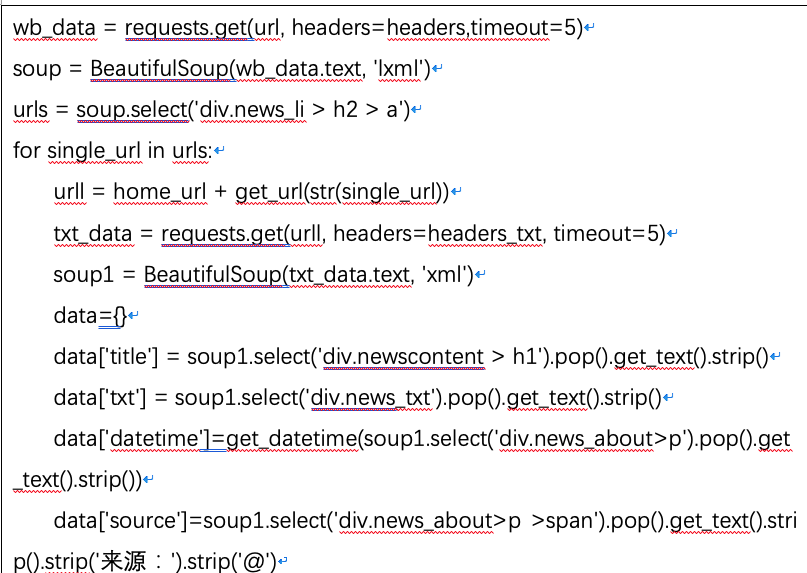


图5.12 新闻抽取核心代码示意图

图5.12代码通过调用requests模块库的get函数请求获取网页资源，运行BeautifulSoup库函数解析网页，得到主新闻网站所包含的所有新闻网址，再通过相同方式获取子新闻详细网页资源，使用select函数获取特定需求信息，采用python字典数据类型存储。

### 5.5.2 新闻存储

采用了csv本地文件+数据仓库双存储方法，核心代码为：

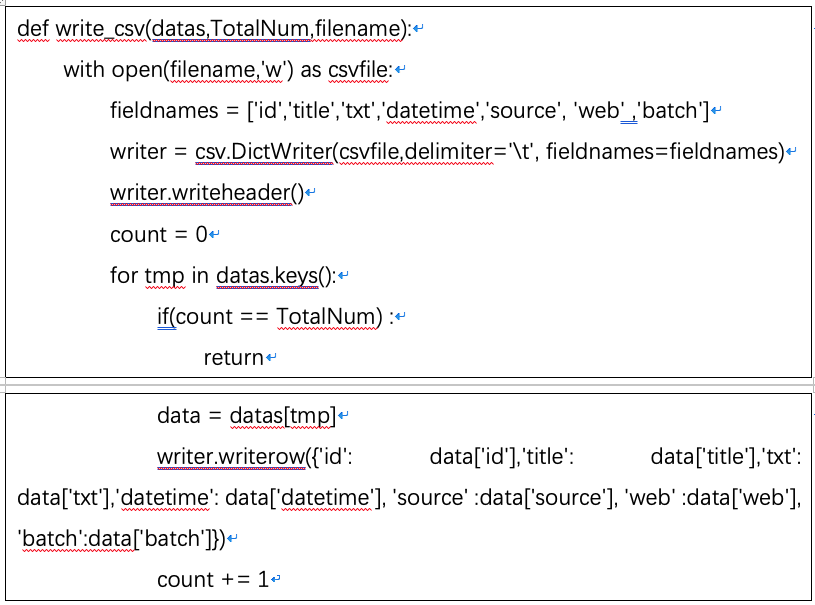


图5.13 新闻存储核心代码示意图

图5.13代码通过调用csv库函数的dictWriter()函数的方式，将新闻资源以字典输入形式输出为csv文件。

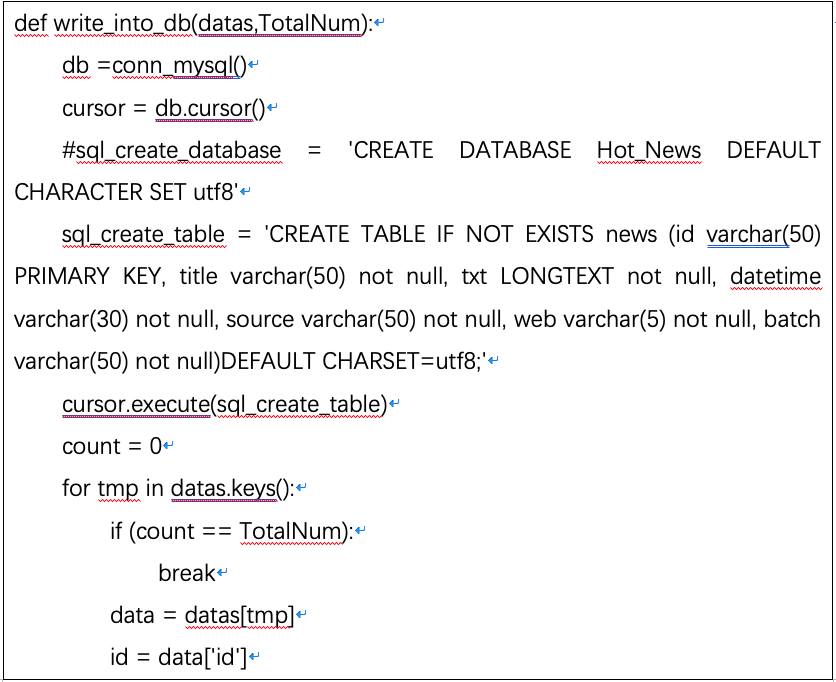


图5.14 新闻存储核心代码示意图

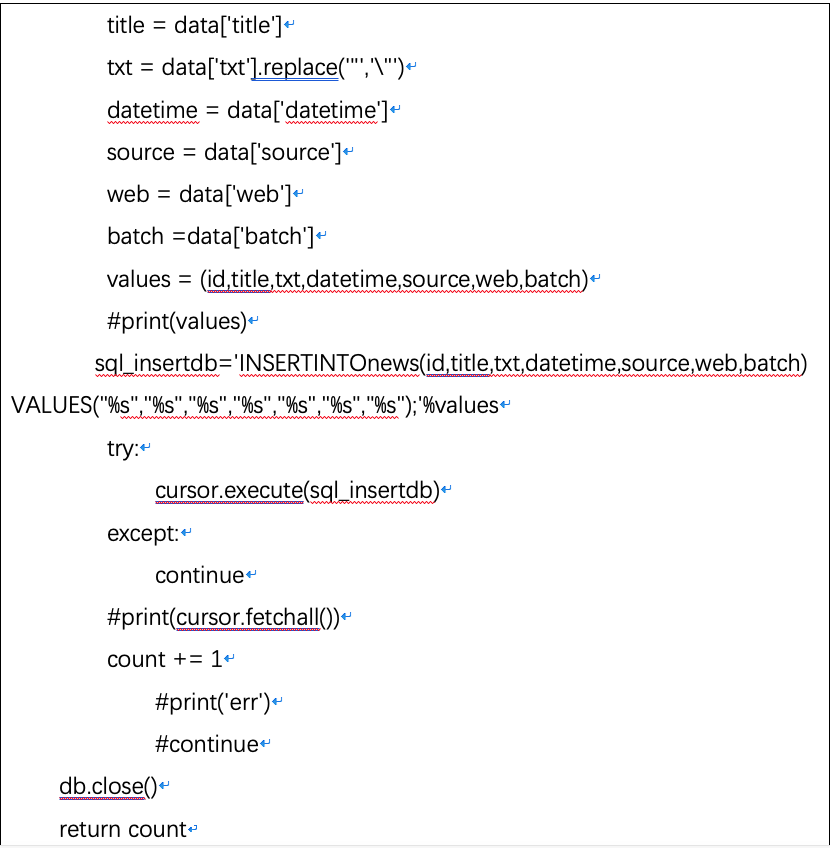


图5.15 新闻存储核心代码示意图

图5.14,5.15代码通过调用pymysql模块库的connect(),cursor(),execute()等函数，与mysql数据库新建连接并按游标运行相应代码返回结果的方式插入新闻抽取函数所获取的新闻资源。

## 5.6 文本聚类模块实现

### 5.6.1 文本向量实现

采用了TF-IDF来实现文本向量的表达，具体代码如下：

图5.16 新闻聚类核心代码示意图

all\_doc = get\_alldoc(data)

all\_corpusdict = get\_doc(all\_doc)

all\_vector = []

for i in range(len(all\_doc)):

vec = get\_vector(all\_doc[i], all\_corpusdict)

all\_vector.append(vec)

all\_vector = np.array(all\_vector)

图5.16代码中函数get\_alldoc()调用了参数data，data表示新闻文本经过预处理之后的数据。函数get\_alldoc()和函数get\_doc()表示得到所有的文本和词袋。函数get\_vector()表示得到全部文本集的向量all\_vector。

### 5.6.2 文本聚类模块的实现

采取了基于K-Means的文本层次凝聚算法，简要代码如下：

cul = []

for i in range(len(all\_vector)):

cul.append(i)

len1 = len(all\_vector)

if len1 < 150:

n = round(math.sqrt(len1))

else:

n = round(len1 / 20)

if n <= 1:

n = 2

K\_cul, score\_K = get\_Kmeans(all\_vector, n, cul)

Agg = []

for i in K\_cul:

m = int(len(i) \* 0.1)

if m <= 1:

Agg.append(i)

else:

cul = get\_Agg(all\_vector[i], m, i)

Agg += cul

score\_Agg = get\_score(all\_vector, Agg)

centroid = []

for i in Agg:

b = np.sum(all\_vector[i], axis=0) / len(i)

centroid.append(b)

cul = []

for j in range(len(Agg)):

cul.append(j)

Log = get\_Agg(centroid, n, cul)

Final\_cul = []

for j in Log:

cul = []

for p in j:

cul = cul + Agg[p]

Final\_cul.append(cul)

score\_end = get\_score(all\_vector, Final\_cul)

if score\_K >= score\_Agg:

score\_max = score\_K

else:

score\_max = score\_Agg

if score\_max >= score\_end:

score\_max = score\_max

else:

score\_max = score\_end

if score\_max == score\_K:

Final = K\_cul

elif score\_max == score\_Agg:

Final = Agg

else:

Final = Final\_cul

图5.17 新闻聚类核心代码示意图

由于是做文本聚类，最好是用余弦距离，而调用常用的的sklearn库中的KMeans模型，默认相似度是欧几里得距离，通过查询找到了biopython库，其中的kcluster模型可以定义相似度计算类型，并且他能运行输入的次数并返回轮廓系数最大的聚类结果。相对应的函数为：

clusterid, error, nfound = kcluster(data, n, dist='u', npass=10)，

这个函数返回的是一个元组 (clusterid, error, nfound) , 其中clusterid是一个每行或则列对应的类的编号；error是最优解的类内的距离和；nfound是最优解被发现的次数。函数包含以下参数：

nclusters (默认:2):类的数目。

k.transpose (默认：0)：选择使用data的行(transpose==0)，或者列(transpose==1)计算距离。

npass (默认:1)：k-means/-medians 聚类算法运行的次数，每次运行使用不同的随机的起始值。如果指定了initialid，npass的值会忽略，并且聚类算法只会运行一次。

method (默认:a)：指定确定聚类中心的方法：method=='a':算数平均值(k-means clustering);method=='m':中间值(k-medians clustering).

当指定method使用其他值时，算法会采用算数平均值。

dist (默认:e'，Euclidean distance)：

'e': Euclidean 距离;

'b': City-block 距离.

'c': Pearson 相关系数;

'a': Pearson相关系数的绝对值;

'u': Uncentered Pearson correlation （相当于两个数据向量的夹角余弦值）

'x': uncentered Pearson correlation的绝对值;

's': Spearman’s 秩相关系数;

'k': Kendall’s τ.

最后选取轮廓系数系数最大的聚类结果，将结果输出。

## 5.7 热点话题抽取与热度值计算模块实现

首先对聚类进行预处理以进行关键词提取和热度值计算，简要代码如下图5.18：

def getmsg():

alltxt=[]

data = pd.read\_csv('./rm\_20200402164450.csv',sep='\t',encoding="utf-8")#读取文件

for i in range(len(cno)):

txt=''

for j in cno[i]:

txt=txt+data.at[j,'txt']+'\n'

alltxt.append(txt)

atime=[]

c=[]

for k in data.index:

row=data.at[k,'datetime']

row=row.replace("年","-").replace("月","-").replace("日","").replace(" ","")

date1=datetime.datetime.strptime(row,'%Y-%m-%d%H:%M').date()

atime.append(date1)

return alltxt,atime,len(data)

图5.18 新闻抽取核心代码示意图

对于话题抽取模块，主要调用的textrank4zh函数，简要代码如下图5.19：

def textrank\_extract(text,keyword\_num):

textrank = analyse.textrank

analyse.set\_stop\_words('stopwords.txt')

keywords = textrank(text, keyword\_num)

word1=""

for keyword in keywords:

word1=word1+(keyword + "/ ")

return word1

图5.19新闻抽取核心代码示意图

然后对处理好的文章进行热度值计算，简要代码如下图5.20：

def calculate(Cno,num,atime):

N=num#所有文章的数量

hot=[]

b=0

for a in range(len(Cno)):#寻找簇内最长时间和最短时间

n=0

MAX=atime[Cno[b][0]]

MIN=atime[Cno[b][0]]

for d in range((len(Cno[b]))):

if MAX<atime[Cno[b][d]]:

MAX=atime[Cno[b][d]]

if MIN>atime[Cno[b][d]]:

MIN=atime[Cno[b][d]]

n=n+1

b=b+1

dec=((MAX-MIN).days)

Hot=((n)/(dec+1))\*(n/N)

hot.append(no)

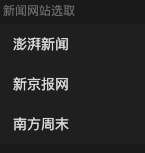
return hot,b

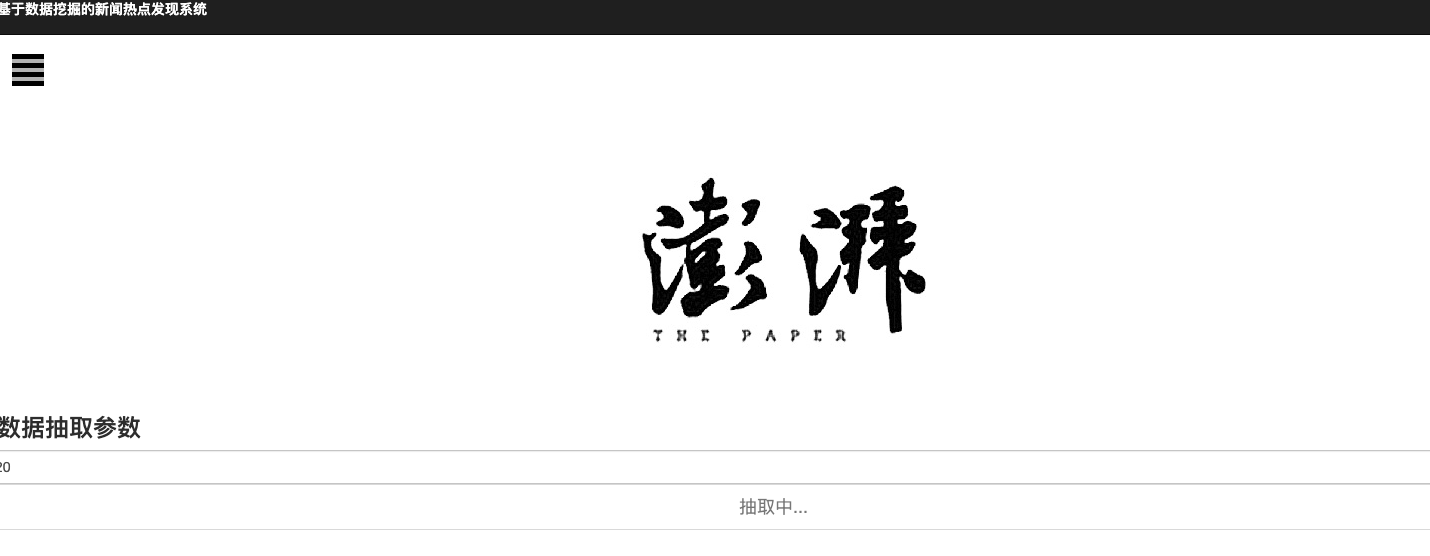
图5.20新闻抽取核心代码示意图

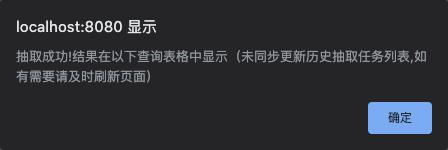
## 5.8 功能性测试与分析

功能性测试包括新闻抽取模块、文本聚类模块、话题抽取模块与历史任务查询模块四个功能进行测试。

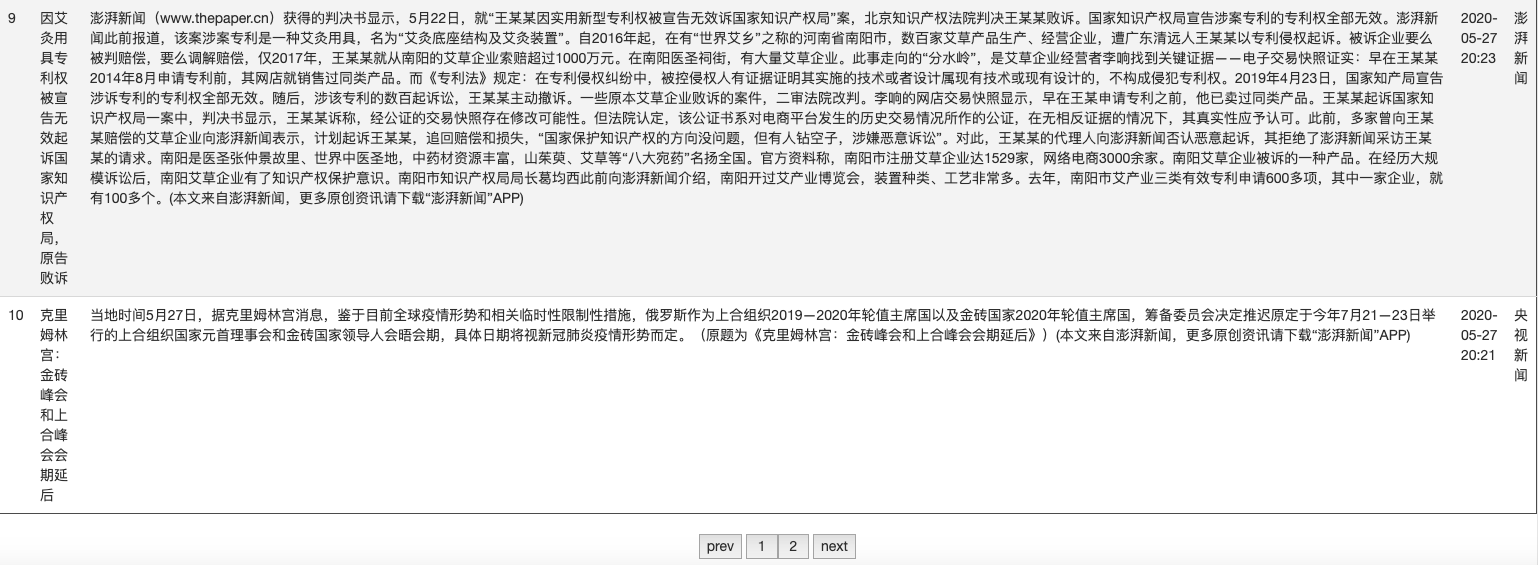
新闻抽取模块测试如下图所示，进入任务页后，首先选取新闻网站，后输入抽取新闻数量来创建抽取任务，任务完成后网站反馈完成任务提醒，并在当前页面下以表格形式展现本次任务抽取的所有新闻（以澎湃新闻抽取20条新闻数据为例）。若任务失败后台报错，网页会有error提醒，任务页重置。







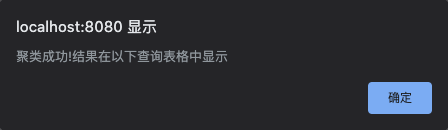


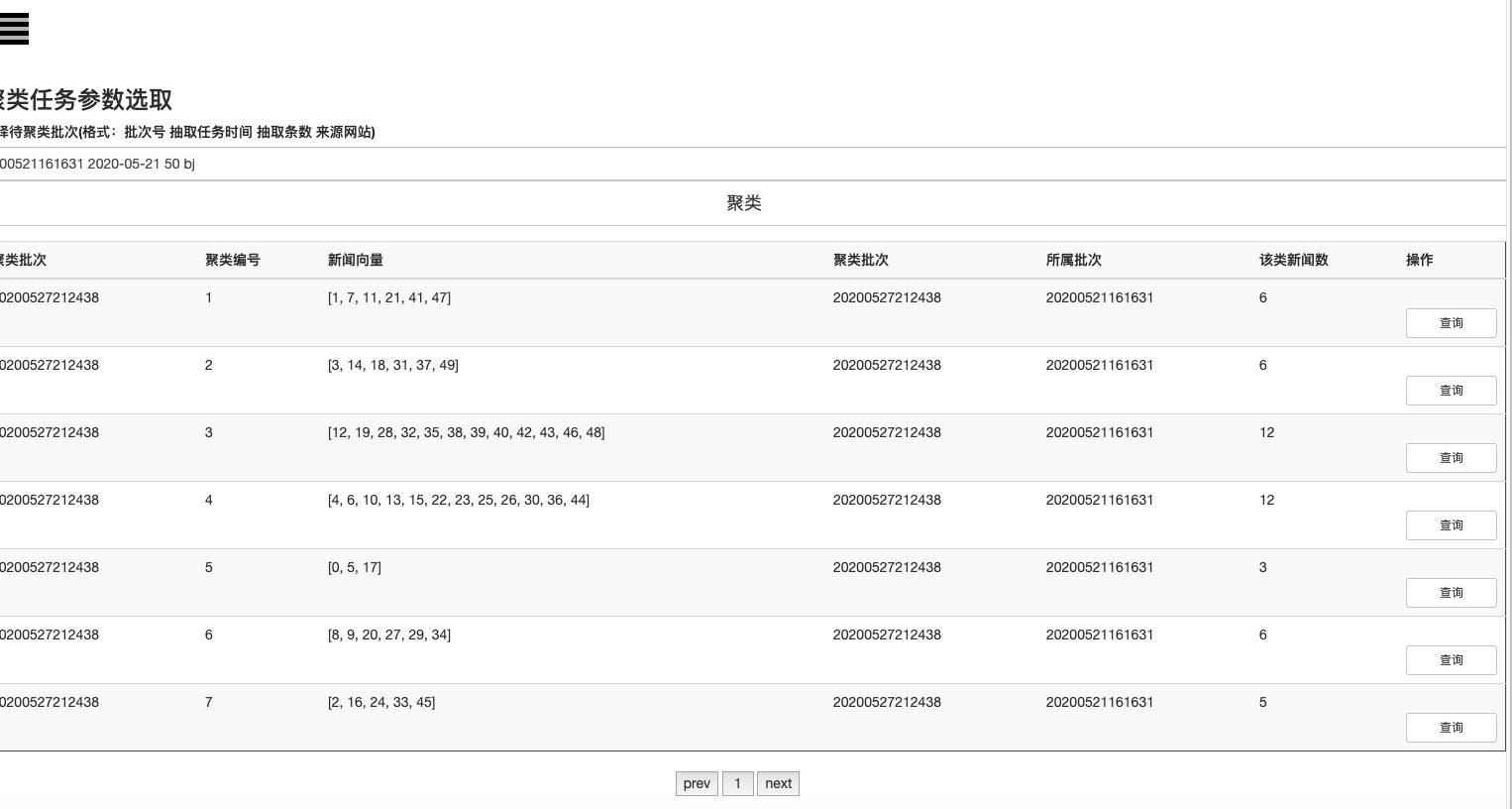


文本聚类模块功能测试主要用于测试聚类功能的可用性。如下图所示（以对20200521161758批次号的50条新闻进行聚类为例），进入功能任务界面，以特定抽取批次的所有新闻作为对象启动聚类任务，聚类对象只能为未进行聚类过的某批次新闻（不能重复对相同批次新闻聚类）。聚类任务完成后网站反馈完成任务提醒，并在当前页面以表格形式展示本批次新闻聚类结果。若任务失败后台报错，网页会有error提醒，任务页重置。





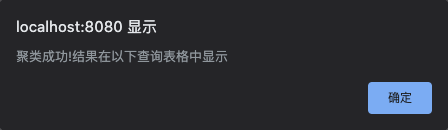




热点话题抽取模块测试如下图所示（用例为20200521160507批次号下的200条来自澎湃新闻的已聚类结果进行话题抽取，关键词数量默认为10为例），话题抽取只能对已聚类的新闻批次进行相关任务启动，可以对相同聚类批次发起提取不同数量关键词的话题抽取任务，对当前批次新闻所有类别基于textrank算法和热度值计算公式分别进行关键词抽取以及热度值计算。话题抽取任务完成后网站反馈完成任务提醒，并在当前页面以表格形式展示本批次话题抽取结果。若任务失败后台报错，网页会有error提醒，任务页重置。









基于上述的功能性测试，文本实现的新闻热点发现系统基本满足预期目标，其中新闻文本获取模块可以高效地实现新闻文本的爬取，文本聚类模块和热点话题抽取模块也基本实现第三、四章设计的算法与模型的相关功能；通过分析最终话题热度和话题排行的相关数据，本系统的所有功能己满足预期目标。

## 5.9 性能测试与分析（轮廓系数）

由于新闻文本聚类没有对应的标准答案，所以只能采用轮廓系数的形式对聚类结果进行评估。

好的聚类：内密外疏，同一个聚类内部的样本要足够密集，不同聚类之间样本要足够疏远。

轮廓系数计算规则：针对样本空间中的一个特定样本，计算它与所在聚类其它样本的平均距离，以及该样本与距离最近的另一个聚类中所有样本的平均距离，该样本的轮廓系数为:

将整个样本空间中所有样本的轮廓系数取算数平均值，作为聚类划分的性能指标。

轮廓系数的区间为：[-1, 1]。 -1代表分类效果差，1代表分类效果好。0代表聚类重叠，没有很好的划分聚类。

实验所用的293条数据划分成了14个簇，所测得轮廓系数结果约为2.56764%，由此可见算法仍有极大进步空间。

../Library/Containers/com.tencent.qq/Data/Library/Caches/Images/9653A0CEE162126CDA0B273CF0A27EC9.jpg

# 第六章 总结与展望

## 6.1 总结

这学期的实践过程中，根据项目计划，我们大概把整个项目分为数据爬取、向量化、聚类和热点话题提取四个模块。

在一个完整的创新实验过程，与以往做一个个独立的简单测验不同，一个完整的科研创新实验，从最初的方案选取，到实施初步计划，到具体实施计划，再到项目的实施、结果分析，都需要我们考虑周全，按部就班地进行。通过自学，完整的把整个系统自行搭建完成，形成最终的系统，并能实际独立运行，爬取数据并通过聚类分析出热点话题。在一开始由于没有经验加之考虑不周到，一开始排出的计划难免出错，后面通过不断改进，随着项目的慢慢发展，在大家的共同努力下，我们的项目才得以顺利进行。团队合作沟通必不可少，出了每周的例行讨论外我们亦会在课余时间不断交流，希望能从中找到新的思路。

在这期间，通过理论知识的自我学习，社会实践的检验、丰富、充实，得以消化吸收，真正变成了自己的知识和经验。避免了只能停留在纸上谈兵的层次上，从而发挥了它的使用价值和社会价值。最深的体会就是要善于勤于思考，主动动手动脑。创新实验不是基础化学课上的实验，只要按着老师讲的步骤做就行了。做的课题对于我们来说，可能是一个没有接触过的新领域，没有人告诉我们一步步该怎么做。需要自己去找文献查资料，去弄明白实验的原理，然后确定要创新的方向。按照这个方向一点点努力，所以每一步都需要独立思考。除此以外，遇到问题时通过讨论并寻找解决问题的方法，并丰富了自己实践经验，是一次不可多得的经历。

初期的选题是针对疫情期间新闻真假进行选择，以此为契机选择了老师提供的新闻爬取和热点抽取的选题，本着为社会提供便利的初心开始了该项目。

在对科学发展观的学习过程中发现，它对我们的思想与科研理念有很大的指导作用。国家以发展为重中之重，科研又以国家的需要与社会的需求为方向，而大学生作为十分宝贵的人才资源，是民族的希望，是祖国的未来。科学发展观以发展为第一要义，指导我们在学习科研知识时不能盲从，要在前人的基础上开拓创新，不断发展和丰富所学知识。加强和改进大学生思想政治教育，提高我们的思想政治素质，把我们培养成中国特色社会主义事业的建设者和接班人，对于全面实施科教兴国和人才强国战略，确保我国在激烈的国际竞争中始终立于不败之地，确保实现全面建设小康社会、加快推进社会主义现代化的宏伟目标，具有重大而深远的战略意义。

在学习上，慢慢地将学习与工作的重点由只是按部就班埋头读书转变到有目标的思考与科研创新中。而在这一学期的学习过程中，自我学习的能力与专业水平都与大一大二相比有所提高，这是令人欣慰的。

## 6.2 系统未来展望

1.尝试在已完成基础上对话题抽取的聚类确定每个类别数量，加入keyhots结果中并能显示；

2.聚类得到的是向量，希望以后能做到点击向量显示该新闻内容(跳出小窗口)；

3.本系统新闻抽取采用requests+beautifulSoup实现手动抽取，以后自动定时抽取应该要改用Scrapy框架；

4.考虑新闻扒取、迭代式分词聚类及关键字提取、热力值计算一键自动化（每天特定时间段自动提取数据）。用户只需选取特定网站扒取特定数量，等待若干时间后可直接查询各类结果。

5.聚类算法从本地抽取特定条目新闻进行聚类，轮廓系数指标效果没达到理想效果，将来打算寻找特定测试集（包含给定相对应的结果簇/簇数），更改评估指标为F-measure,利用混淆矩阵来做。

# 附录 参考文献

【1】基于数据挖掘的网络新闻热点发现系统设计与实现\_童昱强

【2】基于K\_Means的文本层次聚类算法研究\_尉景辉

【3】基于多类别文本的新闻热点发现系统设计与实现\_花道科