

****

**本科毕业论文（设计）**

**题目：**基于web的主题爬虫原型系统

**学生姓名 强佩佩**

**学 号 2013118034**

**指导教师 陈莉**

**院 系 信息学院**

**专 业 软件工程**

**年 级 2013级**

**教务处制**

**二○一七年五月**

诚信声明

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所取得的成果。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或在网上发表的论文。

特此声明。

论文作者签名：

日 期： 年 月 日

**摘 要**

在信息大爆炸的时代，互联网发展迅速，出现了大量的信息载体。通用搜索引擎已不能满足不同领域的人们的搜索需求，垂直搜索引擎的出现是必然的。主题爬虫作为垂直搜索引擎的核心部分，其发展是很重要的。

基于Web的主题爬虫原型系统严格遵循软件开发过程的规范。原型系统采用AngularJs 前端框架、Spring Boot Java框架和WebMagic爬虫框架进行搭建；采用深度优先爬取策略、K近邻分类算法等实现系统功能。本原型系统实现了只爬取新闻类主题的信息，并将主题按照体育、财经、综合、娱乐的主题分类展示在Web页面上。

关键字：主题爬虫，原型系统，Web，K近邻分类算法

**ABSTRACT**

With the rapid development of Internet,it appears a lot of information carriers in the era of information explosion. The general search engine can not meet the needs of people in different areas,so vertical search engine is inevitable. As the core part of the vertical search engine, the development of the topic crawler is very important.

The prototype system of theme crawler based on Web strictly follows the software development process. The prototype system is built by AngularJs front end frame, Spring Boot Java framework and WebMagic crawler framework. And the system functions are realized by using the depth first crawling strategy and K nearest neighbor classification algorithm. The prototype system can only crawl the information of the news theme, and the theme is displayed on the Web page according to the classification of sports, finance, entertainment and entertainment.

**Key words:** topical crawler，protosystem，web, k-nearest neighbor classification

目录

[1绪论 1](#_Toc481519689)

[1.1背景和研究意义 1](#_Toc481519690)

[1.2国内外研究现状 1](#_Toc481519691)

[1.3本文研究内容 1](#_Toc481519692)

[2需求分析 2](#_Toc481519693)

[2.1爬虫概述 2](#_Toc481519694)

[2.2功能需求 2](#_Toc481519695)

[2.3性能需求 3](#_Toc481519696)

[2.3.1时间特性 3](#_Toc481519697)

[2.3.2数据库管理要求 3](#_Toc481519698)

[3总体设计 3](#_Toc481519699)

[3.1总体架构 3](#_Toc481519700)

[3.2模块设计 4](#_Toc481519701)

[3.2.1 web页面模块 4](#_Toc481519702)

[3.2.2爬虫模块 4](#_Toc481519703)

[3.2.3数据库模块 6](#_Toc481519704)

[3.3 类图设计 6](#_Toc481519705)

[4技术实现 7](#_Toc481519706)

[4.1界面实现 7](#_Toc481519707)

[4.2爬虫实现 9](#_Toc481519708)

[4.2.1 Xpath页面解析 9](#_Toc481519709)

[4.2.2 k近邻分类算法 9](#_Toc481519710)

[结论 10](#_Toc481519711)

[参考文献 11](#_Toc481519712)

# 1绪论

## 1.1背景和研究意义

大数据时代，在互联网中有巨量的资源。现有的大型搜索引擎都是综合性的。例如Google、百度、雅虎等都是通用搜索引擎，为所有的用户服务，注重爬取搜索覆盖率，但是却有75%的信息是通用引擎搜索不出来的。所以垂直搜索引擎是顺应时代的必然产物，主题爬虫技术作为垂直搜索引擎的核心部分，只针对某一个主题领域进行信息检索、遍历特定主题的网页，能使用户更快，更准确地检索到更多的需要的信息。所以对主题爬虫的研究显得尤为重要，主题爬虫也将会在很多领域有很大的贡献，比如个性化信息检索[1]、数字图书馆[1]等。

## 1.2国内外研究现状

1993年美国麻省理工学院的学生Matthew Grey写了世界上第一个网络爬虫，并将其命名为“万维网漫游者”[2]。后来，人们将其运用到初期的搜索引擎中。1999年Chakrabarti S等人在 WORLD WIDE WEB大会上第一次提出“主题爬虫”概念[3]。主题爬虫主要使用了主题分析技术，目前已经有很多成熟的算法来支持主题相关度分析，例如D.Harman提出的布尔模型[4]、K近邻分类算法[5]、Salton等人提出的空间向量模型[6]、Vapnik等人的SVM分类算法[7]等。随着主题爬虫算法的成熟，有很多研究团队将主题爬虫应用到搜索引擎中，比如北京化工大学研发了化工领域的搜索引擎。

## 1.3本文研究内容

本文旨在实现一个基于web的主题爬虫原型系统。本主题爬虫原型系统采用深度爬取策略爬取新闻类网页，经过XPath页面解析，运用k近邻分类算法按照不同的主题将爬取的网页分类，例如体育、财经和娱乐等。然后在web页面上按不同的主题展示爬取的网页。

# 2需求分析

## 2.1爬虫概述

网络爬虫是一个自动爬取网页的程序。根据所给的初始URL种子，按照一定的规则自动抓取网页，在抓取页面的过程中还会从当前页面上不断发现新的URL，然后存放到URL待爬取队列中，直到满足爬取的结束条件。网络爬虫一般分为：通用爬虫[8]和主题爬虫[9]。

通用爬虫的目标是爬取尽可能多的网页，在这个过程中需要耗费很多的系统和网络资源。但是其爬取的网页的主题之间并没有太多的相关性，爬取到的页面的利用率不高。

主题爬虫在爬取网页之前确定一个或多个主题，在爬取时只保留与选定主题相关的爬取结果，判断URL队列，进行下一次的爬取，再获取URL，直到达到结束条件。相对通用爬虫来说，主题爬虫提高了WEB资源中主题资源覆盖率[10]。

基于web的主题爬虫原型系统旨在互联网上爬取新闻网页，经过主题分析之后，按照不同的主题分类将新闻网页爬取结果展示在WEB页面上。

## 2.2功能需求

基于WEB的主题爬虫原型系统由三个模块组成，分别是爬虫模块、数据库模块和Web页面展示模块。

1. 爬虫模块：根据给定的种子，按照预设的深度策略通过http协议爬取web网页，进行网页解析，提取页面有用信息。在这个过程中，发现新的URL放入待爬取种子队列中，并对其进行主题分析，将爬取的链接按主题保存在MySQL数据库中。
2. 数据库模块：按主题分类存储主题爬虫模块爬取的经过主题分析之后与主题相关的网页链接以及摘要。
3. Web页面展示模块：页面上显示有体育、财经、综合以及娱乐四个主题。根据用户的选择，将保存在数据库中的链接以及摘要按主题展示在Web页面上。

## 2.3性能需求

### 2.3.1时间特性

1. 响应时间应该在用户的视觉感知范围内，一般为3秒。
2. 为避免多个用户同时使用系统时，过多线程使返回结果慢，本系统采用了定时任务，每三个小时执行一次，使响应时间得到保障。

### 2.3.2数据库性能要求

本系统采用MySQL数据库存储数据，而爬虫系统的运行时间越长爬取结果越来越多，用建立主键索引的方式在海量数据下依然让系统具有很低的响应延迟，保证了服务的流畅性。

# 3总体设计

## 3.1总体架构

主题爬虫的总体架构如图3.1所示。

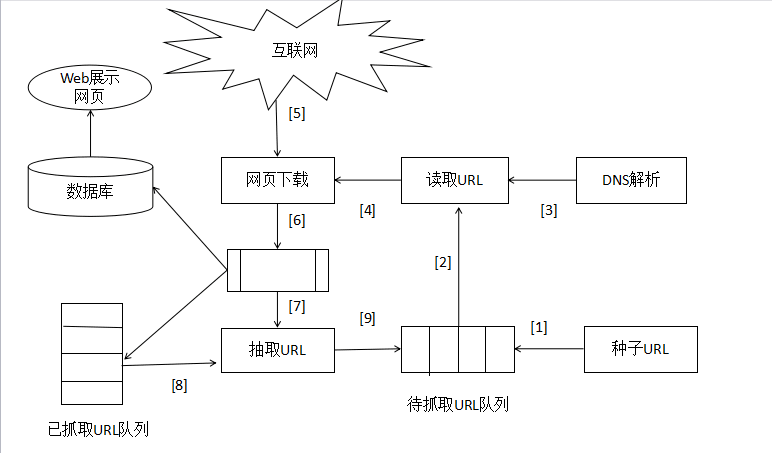


图3.1 总体架构图

1. 首先选定种子URL，将种子URL放到待抓取队列中。
2. 从待取URL队列中依次读取待抓取URL。
3. 将URL通过DNS解析，把链接地址转换为网站服务器对应的IP地址。
4. 将其和网页相对路径名称交给网页下载器。
5. 网页下载模块下载URL的网页。
6. 分为两个部分操作：一是将下载的网页的URL保存在已抓取URL队列；二是将下载下来的网页进行主题分析，与主题相关的页面的URL保存到数据库，与主题无关的舍弃。
7. 抽取主题相关的网页的URL。
8. 对比已抓取URL队列，查看URL是否被抓取过，如果没有被抓过取，将其放入待抓取URL队列中。

## 3.2模块设计

### 3.2.1 web页面模块

Web页面上展示四个可选主题，分别为体育、财经、综合、娱乐，用户根据喜好选择主题，然后在页面下方按列表的方式展示数据库中存放的主题链接及摘要。

### 3.2.2爬虫模块

爬虫模块采用WebMagic框架，模块化设计。

1. Spider组件是整个爬虫框架的调动程序，scheduler是URL管理部分，在不断的发现新的URL，放到URL队列之前进行去重。
2. download组件是页面下载，也是很重要的一部分，在此框架里采用深度优先策略爬取，设置爬取深度为两层。在此页面下载模块设置了Quartz定时任务，每三个小时执行爬取一次。
3. pageprocessor组件是爬虫的核心，它的功能是进行链接提取和页面分析。
4. pileline组件是离线分析和数据持久化，本系统在此部分加入了K近邻分类算法，分析主题相关度。

爬虫模块流程图见图3.2。

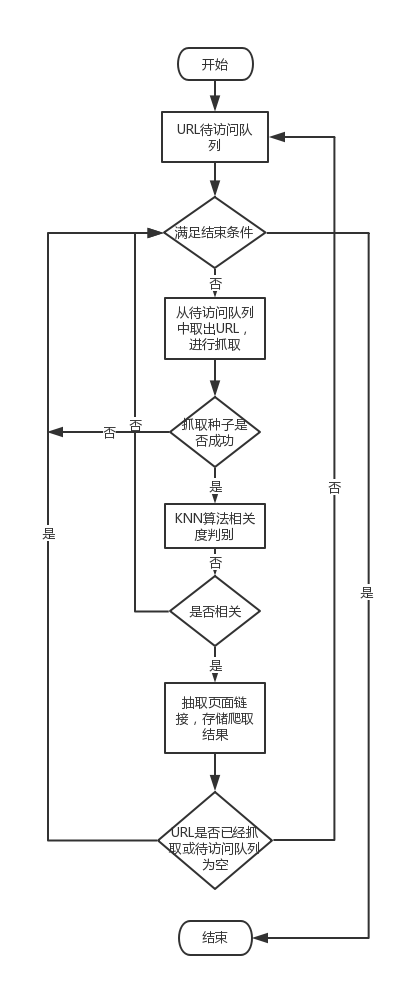


图3.2 流程图

### 3.2.3数据库模块

字段说明：ID是爬取数据的唯一性标示

URL是爬取的链接

Digest是爬取的网页的新闻摘要

Theme是爬取的网页的主题，对应页面上的button标签

把爬取的数据存放到数据库result表，详情见表3.1。

表3.1 result表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 备注 |
| Id | bigint | 20 | 主键 |
| url | varchar | 256 | 非空 |
| digest | varchar | 256 | 非空 |
| theme | varchar | 20 | 非空 |

## 3.3 类图设计

1. Spider类是爬虫模块的调动程序，它向Httpclientdownloader类传递爬取任务的URL List数组，爬取页面。
2. Httpclientdownloader类通过http协议发送Requst请求爬取页面响应，并且把爬取数据保存到实体类Page。
3. PageProcesser接口类，有三个部分：爬虫配置、页面元素提取、发现链接。
4. 爬虫配置部分通过设置配置类Site然后调用接口的setSite()方法对爬虫进行基本配置，包括爬取数据的编码设置、抓取任务调度间隔、爬取任务超时时间、以及爬取失败后重试次数等。
5. 页面元素提取逻辑部分通过接口方法processer()来具体实现，这其中可以使用Page类预留接口的Xpath技术，CSS选择器，正则表达式，JsonPath等技术来进行页面提取。
6. 链接发现部分通过深度优先策略搜索NEXT链接，以及通过正则匹配对链接做基本排除的双重模式。
7. pipeline接口类，用于数据离线处理和持久化，其中FilePipeline实现类用来对数据的本地文件保存以供分类器模块ClassiFierUtil类对爬取数据分类，MySqlPipeline实现类将ClassiFierUtil分类完成的内容持久化到MySQL数据库中。

爬虫模块类图设计如图3.3。

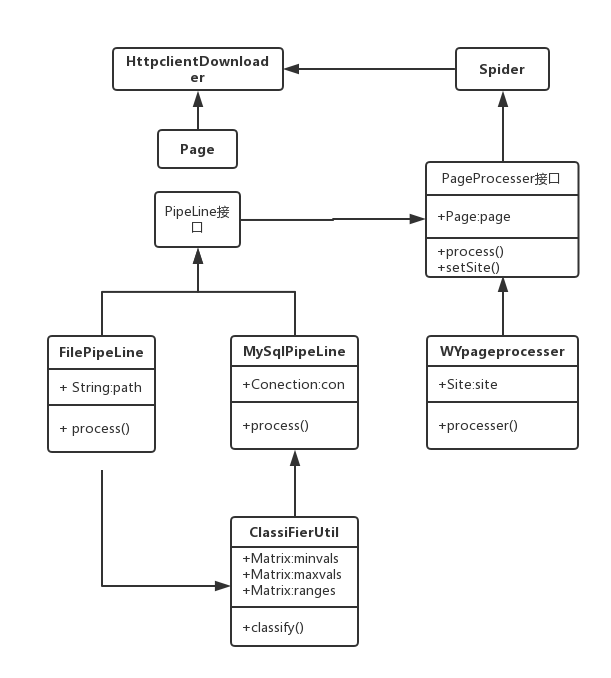


图3.3 类图

# 4技术实现

## 4.1界面实现

本系统WEB界面实现运用了Bootstrap框架以及AngularJs框架。AngularJs是一个前端MVC WEB框架，扩展并增强了HTML功能[11]。Bootstrap框架，是CSS[12]、HTML框架。

页面展示用单页面简约设计，通过前端AngularJS框架异步Http请求API获取爬取结果，代码见下图4.1：



图4.1

用Bootstrap渲染界面效果，框架ng- repeat属性进行数据双向绑定~~，核心代码：~~

~~<table class="table" ng-hide="tableState">~~

~~<thead>~~

~~<tr>~~

~~<th>~~

~~编号~~

~~</th>~~

~~<th>~~

~~URL~~

~~</th>~~

~~<th>~~

~~摘要~~

~~</th>~~

~~<th>~~

~~主题~~

~~</th>~~

~~</tr>~~

~~</thead>~~

~~<tbody>~~

~~<tr class="success" ng-repeat="x in results">~~

~~<td> {{x.id}}</td>~~

~~<td> {{x.url}}</td>~~

~~<td> {{x.digest}}</td>~~

~~<td> {{x.theme}}</td>~~

~~</tr>~~

~~</tbody>~~

~~</table>~~

界面上展示的是爬下来的链接以及新闻title，将它们按主题分组展示。 进入爬虫网页界面效果见下页图4.1。

用户选定主题，将已经爬取并且分析完存储在数据库里的数据展示在页面上，见下页图4.2。



图4.1 主题爬虫主界面

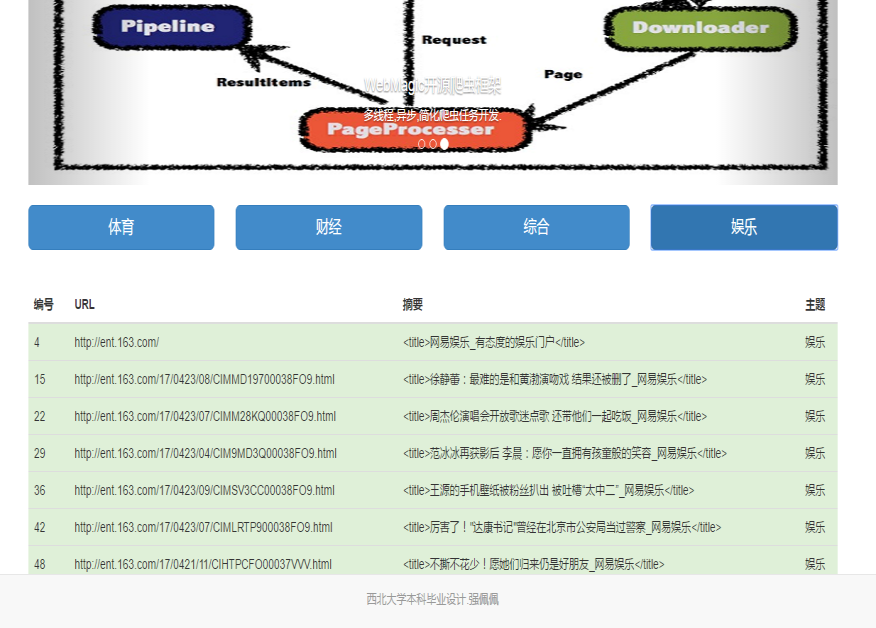


图4.2娱乐主题新闻展示图

## 4.2爬虫实现

### 4.2.1 Xpath页面解析实现

Xpath页面文本提取的实现主要通过扩展Jsoup框架来完成。ElementSelector定义了html文本的提取方法，Selector定义了text文本提取的基本方法,BaseElementSelector类Implement了ElementSelector和Selector并且扩展Jsoup框架实现文本的提取,。XpathSelector继承BaseElementSelector用Xpath语言对页面文本进行提取。也可以通过继承BaseElementSelector类用CSS选择器、正则表达式的方式来对页面文本进行提取。

类结构如图4.3。

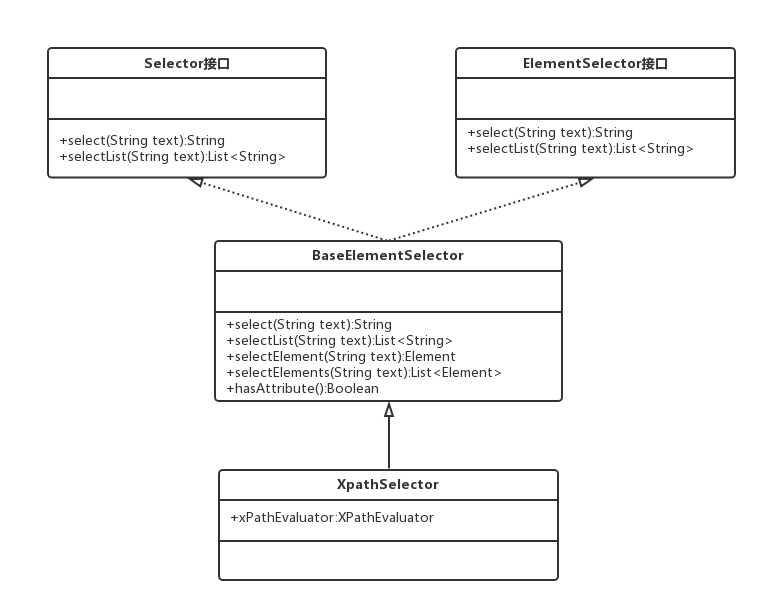
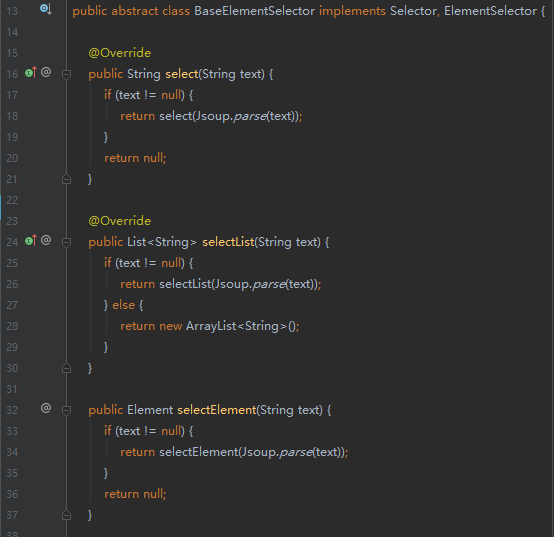


图4.3 类图

。

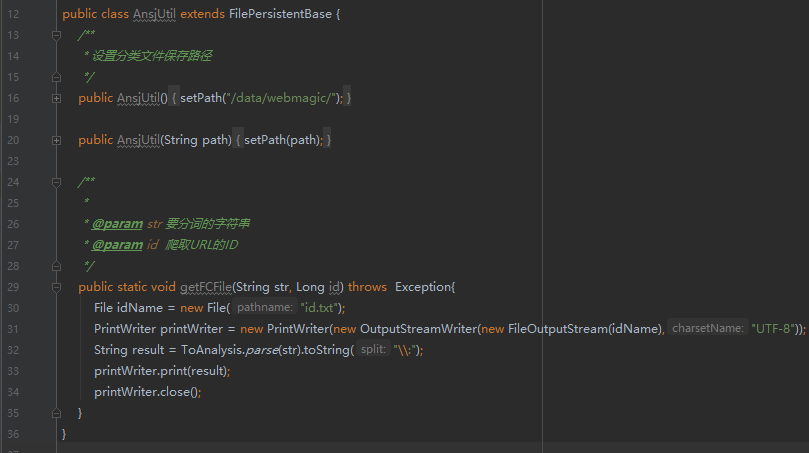
代码如图4.：



### 4.2.2 分类器实现

1. 分类器的基础分词系统实现

分词系统实现使用开源Ansj中文分词系统，对爬取的新闻页面抽取其标题和正文内容通过分词系统分词后保存为本地文件，供WordsVector类对文档进行向量化处理。爬虫模块在爬取URL时会对每一个爬取的URL设置唯一ID，分词后的本地文件名为URL唯一ID。分词工具类如图4.：



1. KNN算法实现

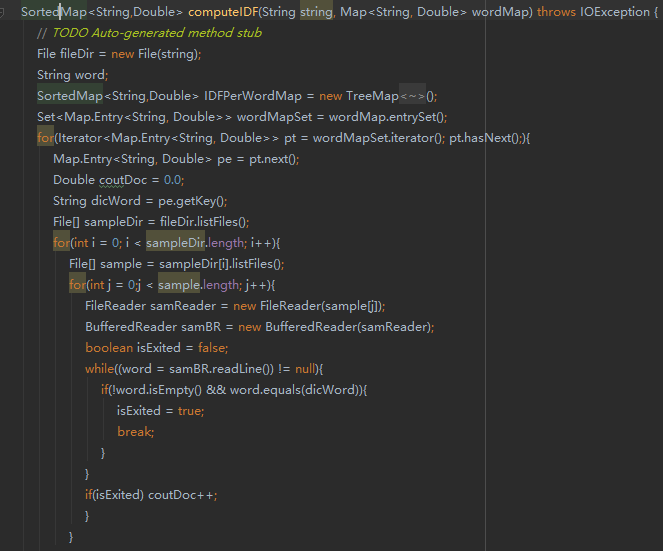
k近邻分类算法(k-nearest neighbor classification)的指导思想是 “近朱者赤近墨者黑”，根据自己的邻居来判定自己的分类[15]。算法的基本方法是：给定一个已经分类的训练集，并且用特征值表示，将要测试的对象也用特征值表示，然后根据测试测试集和各个训练集的特征值之间的距离进行分类。

类WordsVector实现文档向量化。采取对文档进行TF-IDF[16]计算的方法，TF-IDF的计算公式。

（4.1）

（4.2）

核心方法如图：



类ClassifierUtil实现了KNN算法分类的具体逻辑，对待分类URL文本计算相似度后取前5个样本对其所属类别计算权重。得到最大权重的类别，待分类文本就归到该类别下，通过MysqlPipeLine类分类别保存到数据库。本文实现KNN算法的训练集和测试集为手动下载网易新闻后对其内容向量化所建立的。核心代码如图4.5：

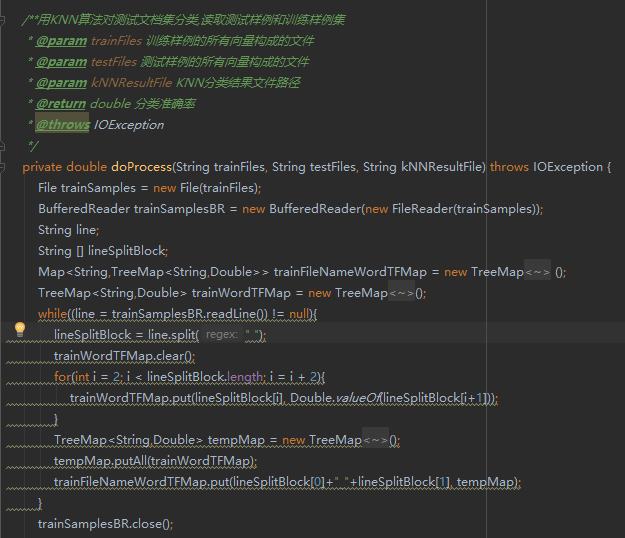


图4.5 KNN算法核心代码

# 5结论

基于JAVA开源爬虫框架，二次开发后实现了从网易新闻网上爬取特定护体新闻的功能,并将其以web页面的形式展示。在系统实现过程爬虫模块采用的webmagic框架，主题分析技术用到的K近邻分类算法等。本系统的功能实现，但是在页面设计上灵活性略有不足，没有提供分页功能，用户浏览不够方便。为此，还需要进一步完善系统，为用户带去更愉快的体验。

# 参考文献

[1] 陈竹敏. 面向垂直搜索引擎的主题爬行技术研究[D]. 山东大学, 2008.

[2] 张俊兰, 刘翼, 铁宏军. 基于CSS技术的网页设计应用研究[J]. 延安大学学报:自然科学版, 2010, 29(3):34-39.

[3] Chakrabarti S, Martin V D B, Dom B. Focused crawling: a new approach to topic-specific Web resource discovery[J]. Computer Networks, 2000, 31(11-16):1623-1640.

[4] Harman D. User-friendly systems instead of user-friendly front-ends[J]. Journal of the Association for Information Science and Technology, 1992, 43(2):164-174.

[5] Cover T M, Hart P E. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1):21-27.

[6]Salton G, Wong A, Yang C S. A vector space model for automatic indexing[M]// Readings in information retrieval. Morgan Kaufmann Publishers Inc. 1997:273-280.

[7] 李正文. 基于SVM分类算法的主题爬虫研究[D]. 哈尔滨工程大学, 2011.

[8] 丁发梅. 一种改进Best-First算法的主题爬虫搜索算法研究[D]. 重庆大学, 2015.

[9]刘金红, 陆余良. 主题网络爬虫研究综述[J]. 计算机应用研究, 2007, 24(10):26-29.

[10] 陈丛丛. 主题爬虫搜索策略研究[D]. 山东大学, 2009.

[11]常红要, 朱征宇, 陈烨,等. 基于HTML标记用途分析的网页正文提取技术[J]. 计算机工程与设计, 2010, 31(24):5187-5191.

[12] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communications of the Acm, 1975, 18(11):613--620.

[13] Tajima K, Fukui Y. Answering xpath queries over networks by sending minimal views[C]// (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004. DBLP, 2004:48-59.

[14] Abiteboul S, Bonifati A, Manolescu I, et al. Dynamic XML documents with distribution and replication[C]// ACM SIGMOD International Conference on Management of Data, San Diego, California, Usa, June. DBLP, 2003:527-538.

[15] 刘汉兴, 刘财兴. 主题爬虫的搜索策略研究[J]. 计算机工程与设计, 2008, 29(12):3160-3162.

[16] 路永和, 李焰锋. 改进TF-IDF算法的文本特征项权值计算方法[J]. 图书情报工作, 2013, 57(3):90-95.