



**3273100**

互联网搜索引擎

项目报告

**题目 基于Java的互联网搜索引擎**

**学院 信息科学与技术**

**教师 吴晓**

**班级 软件2班**

**姓名 高国彬**

**学号 2016112462**

**2018-2019（2）**

目录

[1. 项目要求 1](#_Toc6750747)

[2. 项目步骤 1](#_Toc6750748)

[2.1 爬取网页 1](#_Toc6750749)

[2.2 批量改名 4](#_Toc6750750)

[2.3 特殊符号的去除 5](#_Toc6750751)

[2.4 中文分词及删掉停用词 6](#_Toc6750752)

[2.5 英文分词与删除停用词 8](#_Toc6750753)

[3. 代码 10](#_Toc6750754)

# 1. 项目要求

通过下载引擎（Web Crawler/Spider）自动下载至少500个英文文档/网页，以及500个中文文档/网页，并保留原始的文档/网页备份（如：News \_1\_Org.txt）

编程对所下载文档进行自动预处理：

* 将各个单词进行字符化
* 完成删除特殊字符等操作调研并选择合适的中文分词技术和工具实现中文分词，可参看《Lucene分词器比较》：https://teniteve.com/blog/223396
* 删除英文停用词（Stop Word）
* 删除中文停用词调用或者编程实现Porter Stemming
* 功能将中文文档进行字符化，即可被搜索引擎索引的字符单元
* 对于英文文档，经过以上处理之后，将经过处理之后所形成简化文档保存（如：News\_1\_E txt），以备以后的索引处理
* 对于中文文档，经过以上处理之后，将经过处理之后所形成简化文档保存（如：News\_1\_C.txt），以备以后的索引处理

# 2. 项目步骤

## 2.1 爬取网页

首先是爬取的工具的选取

一开始候选的有Python，java，以及现成的工具

经过一阵比较之后，我选取了java。

原因为

* Python太久没有使用了，对Beautiful库不熟悉
* 后续开发打算使用Java的Swing

抓取的目标

* 爬取的目标一开始选取的知乎。但是知乎一次并且爬取的答案过多，所以考虑更换平台。
* 之后选取了简书，因为简书是典型的博客。但是简书一次只能爬取5篇，并且首页更新不频繁。3小时只爬取了几十个。
* 还考虑了头条，但是头条使用的Ajax异步加载，使用传统的COM抓取，无法获得信息。
* 最后选择了CSDN作为我们抓取的目标。

给我们的爬虫设定了每次抓取之后，等待一秒钟，抓到500个页面即停止。

一晚上过去了，只抓到了378个页面。

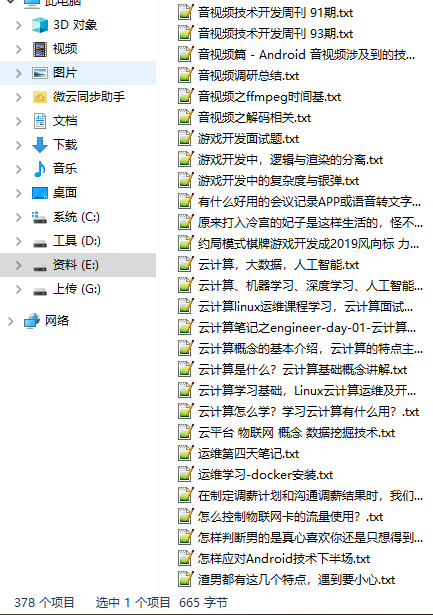


图 1

这就让人有些纳闷了。

后来看了输出，发现反反复复抓取的就是那些页面。

所以应该是CSDN首页推荐的原因，首页推荐的总是这些页面。那这个不行了，剩下的总还是要抓的。

首先我选择了一个作文网。配置完成之后依然不能抓取，这让我很纳闷，最后看了下这个网页的后缀，是shtml，搜索之后发现这个格式不能用这种方式抓取。

于是搜索到了另一个使用html后缀的作文网进行爬取。

先是爬取了首页的作文。然后发现还是不够。

最后是进入到首页的导航中，如一年级，二年级。然后再爬取该区域的作文。这次爬取的作文数目就够了。

然后再手动删除一些失效的文件。

如

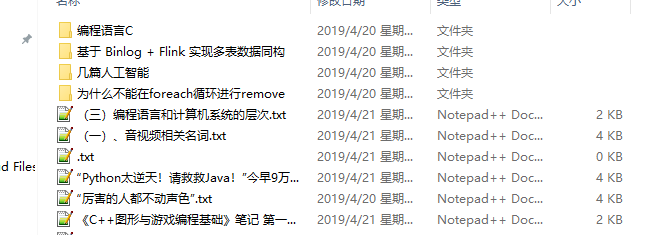


图 2

最后得到中文网页561个

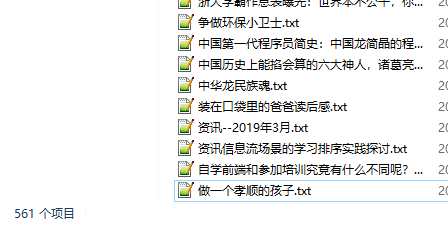


图 3

然后是英文网页。

英文网页选择了一个英文作文网进行爬取。

会时不时出现标题错误，然后爬虫重新爬取这样的事情。然后选择了一个正则表达式匹配正确的文件名格式。之后就顺风顺水，爬取了539个英文网页。

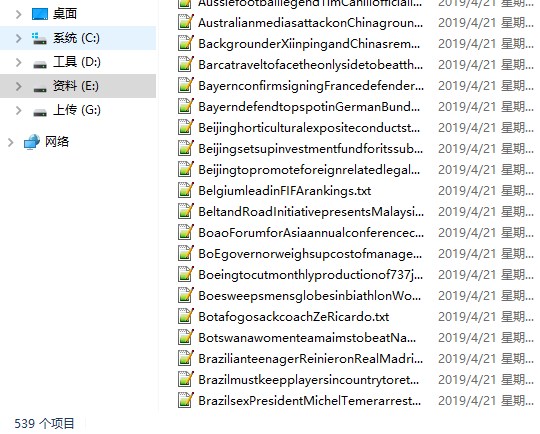


图 4

## 2.2 批量改名

一开始的网页是使用的网页标题进行的命名，现在我们要把这些标题标准化。

发现有比较简单的方法。将文件夹中的文件全选->重命名->填写自己想要的文件名。系统会自动帮忙重命名。

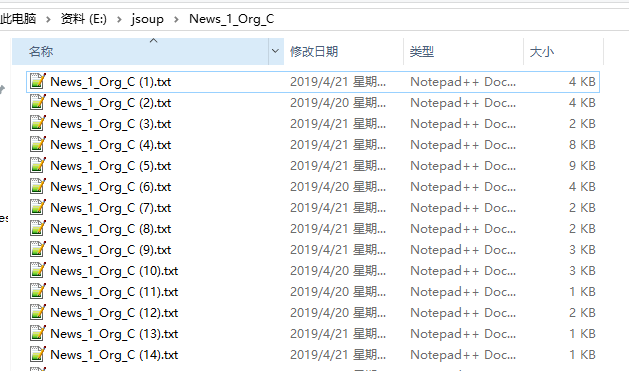


图 5

## 2.3 特殊符号的去除

第一步，使用将所有字符读入。存储到一个字符串中。

第二步，使用正则表达式，将所有特殊符号转换为空格，将连续的空格处理为一个空格。

第三步，使用正则表达式，将中文字符中的非中文字符转换为空格，将连续的空格处理为一个空格。对于英文来说，将英文字符中的非英文字符转换为空格，将连续的空格处理为一个空格

下面为处理之前和处理之后的对比

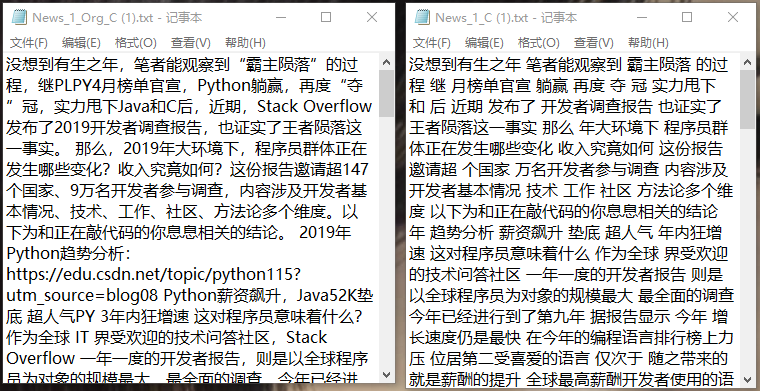


图 6

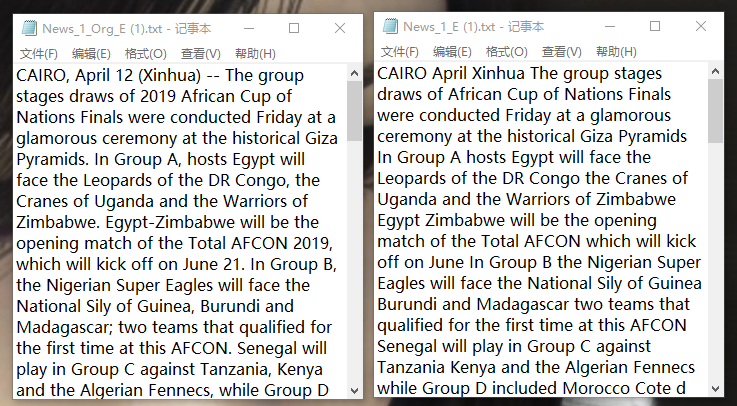


图 7

## 2.4 中文分词及删掉停用词

中文分词调研结果如下图所示

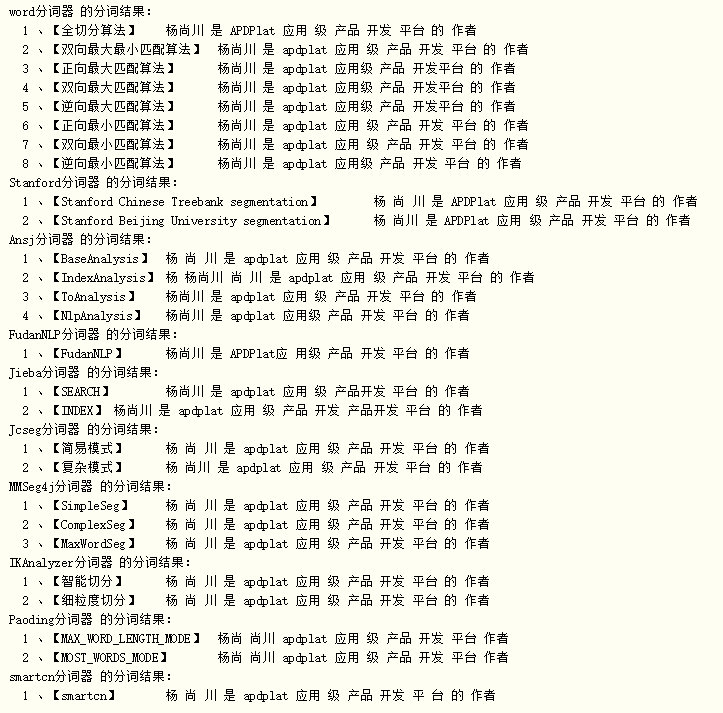


图 8

可以看到，分词最为准确的是word分词器的双向最大匹配算法和逆向最大匹配算法。

我们前往github上，下载word已经编译好的jar包，下载1.3.1版本。

运行的时候报错。提示。

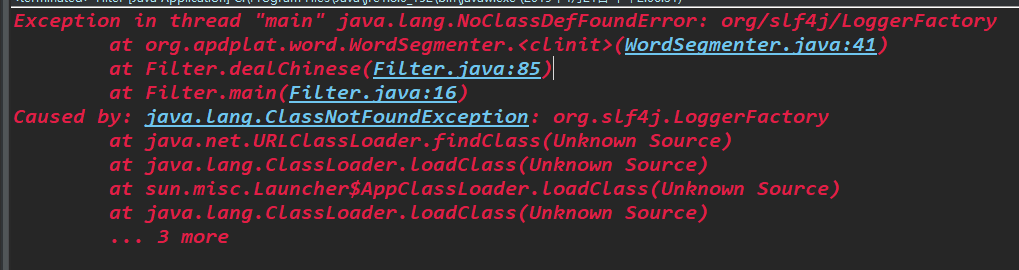


图 9

查阅资料之后发现是因为没有slf4j这个记录日志的包。

网上下载该包，导入到工程中。

slf4j-api-1.7.9.jar

slf4j-log4j12-1.7.9.jar

仍然报错，最后发现是要将

slf4j-log4j12-1.7.9.jar

剔除掉，才能继续运行。

运行的时候虽然有抛出错误，但是能够继续运行。

比较一下中文分词的效果。

并且我们选择了自带的删除停用词的功能，所以两个步骤一起完成了。

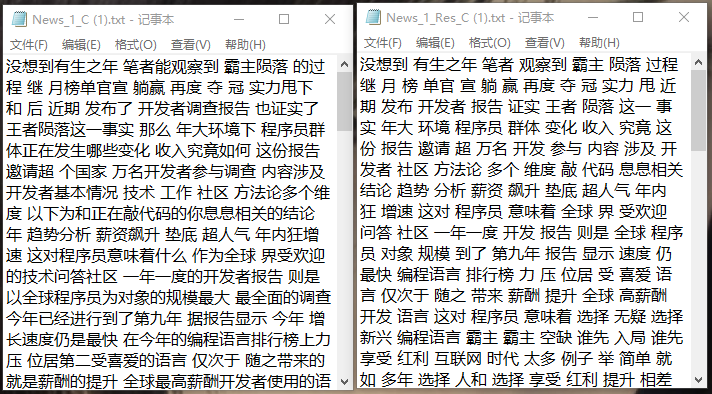


图 10

## 2.5 英文分词与删除停用词

英文分词，我们搜索了一下。

一个博主推荐了Porter stemmer，顺便做了以下的说明。

Porter stemmer 并不是要把单词变为规范的那种原来的样子，它只是把很多基于这个单词的变种变为某一种形式！换句话说，它不能保证还原到单词的原本，也就是"created"不一定能还原到"create"，但却可以使"create" 和 "created" ，都得到"creat" ！

于是，我们获得了源代码，直接新建一个类，复制粘贴。

调用样例里面的方法进行分词。最终得到分词结果。

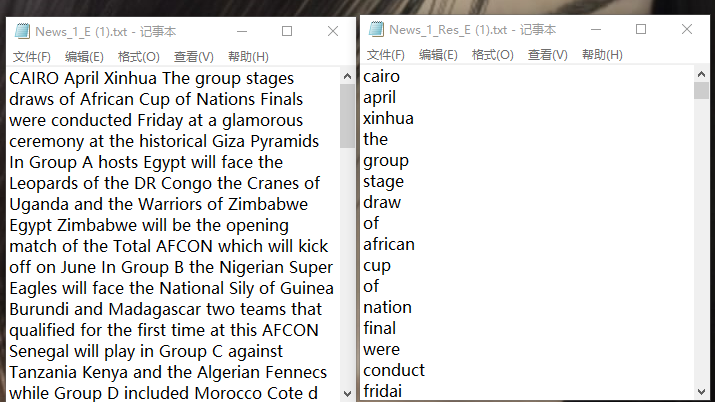


图 11

可以看到，分词完成。但是此时仍有停用词如of和were这种

于是我们需要删除停用词。

首先，我们取查找停用词表。将此停用词表保存为stopWord.txt中。然后将此停用词表按行读取，保存在一个hashSet中。每次分词之后，将分词的结果判断一下，时候存在在hashSet中，如果不在，才写在文件中。

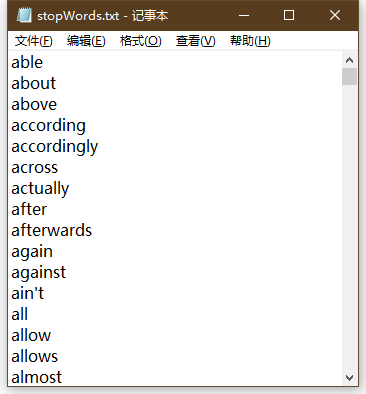


图 12

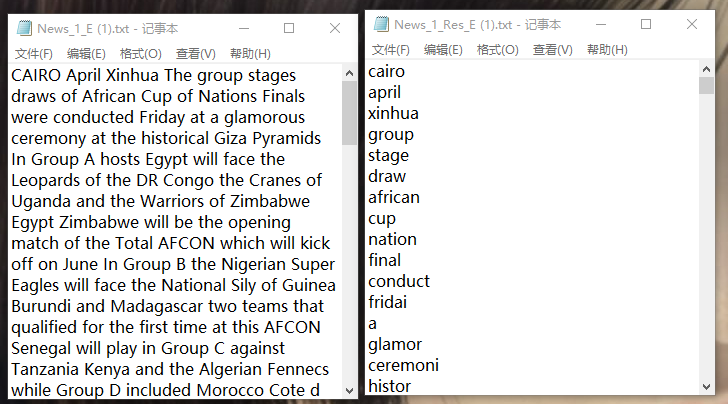


图 13

可以看到，最终得到的英文分词结果是没有停用词的了。

# 3. 代码

import java.io.File;

import java.io.FileWriter;

import java.io.Writer;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

// 爬虫代码

public class Spider {

static int i;

public static void main(String[] arg) {

i = 0;

Spider spider = new Spider();

while (i <= 500) {

try {

// spider.testcsdn();

spider.spiderArticleE();

System.out.println(i);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public void spiderArticleE() throws Exception {

Document document = Jsoup.connect("http://www.hxen.com/englishnews/")

// 模拟火狐浏览器

// .userAgent("Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)")

.userAgent("Mozilla").get();

Elements url = document.select("#content").select("div").select("div.w690.fl").select("div").select("div")

.select("div.pagecon.tc.clearfix.ptb25").select("div").select("select").select("option");

// #content > div > div.w690.fl > div > div > div.pagecon.tc.clearfix.ptb25 >

// div > select > option:nth-child(1)

for (Element question : url) {

String URL = question.attr("abs:value");

System.out.println(URL);

Document document2 = Jsoup.connect(URL).userAgent("Mozilla").get();

Elements articles = document2.select("#content").select("div").select("div.w690.fl").select("div")

.select("div").select("div.newsRankBar.mb20").select("div").select("ul").select("li").select("div")

.select("div").select("h3").select("a");

// #content > div > div.w690.fl > div > div > div.newsRankBar.mb20 > div > div >

// ul > li:nth-child(1) > div > div > h3 > a

for (Element article : articles) {

String URL2 = article.attr("abs:href");

Document document3 = Jsoup.connect(URL2).userAgent("Mozilla").get();

Elements title = document3.select("#slistl").select("div.arcinfo.center").select("b").select("h1");

// #slistl > div.arcinfo.center > b > h1

Elements answer = document3.select("#arctext").select("p");

// #arctext > p:nth-child(2)

System.out.println("\n" + "链接：" + URL2 + "\n" + "标题：" + title.text() + "\n" + "回答：" + answer.text());

String titleString = title.text().replaceAll("[^0-9a-zA-Z]J\*", "");

String filepath = "E:" + File.separator + "jsoup" + File.separator + "word" + File.separator

+ titleString + ".txt";

System.out.println(filepath);

File file = new File(filepath);

if (!file.getParentFile().exists()) {

file.getParentFile().mkdirs();

i++;

System.out.println(i);

}

Writer out = new FileWriter(file);

out.write(answer.text());

out.close();

}

}

try {

Thread.sleep(1000);

} catch (Exception e) {

System.exit(0);// 退出程序

}

}

public void spiderArticleC() throws Exception {

Document document = Jsoup.connect("http://www.zuowenwang.net/")

// 模拟火狐浏览器

// .userAgent("Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)")

.userAgent("Mozilla").get();

Elements url = document.select("body").select("div.menu\_nz").select("dl.menu\_nz\_one").select("dd").select("a");

// body > div.menu\_nz > dl.menu\_nz\_one > dd:nth-child(2) > a:nth-child(1)

for (Element question : url) {

String URL = question.attr("abs:href");

Document document2 = Jsoup.connect(URL).userAgent("Mozilla").get();

Elements articles = document2.select("body").select("div.wrap.wrap3").select("div").select("div.left")

.select("div").select("ul").select("li").select("h3").select("a");

// body > div.wrap.wrap3 > div > div.left > div > ul > li:nth-child(1) > h3 > a

for (Element article : articles) {

String URL2 = article.attr("abs:href");

Document document3 = Jsoup.connect(URL2).userAgent("Mozilla").get();

Elements title = document3.select("#tm\_wrap3").select("div").select("div.left").select("div")

.select("div.article-t").select("h1");

// #tm\_wrap3 > div > div.left > div > div.article-t > h1

Elements answer = document3.select("#tm\_wrap3").select("div").select("div.left").select("div")

.select(" div.content").select("p");

// #tm\_wrap3 > div > div.left > div > div.content > p:nth-child(3)

System.out.println("\n" + "链接：" + URL + "\n" + "标题：" + title.text() + "\n" + "回答：" + answer.text());

String filepath = "E:" + File.separator + "jsoup" + File.separator + "word" + File.separator

+ title.text() + ".txt";

System.out.println(filepath);

File file = new File(filepath);

if (!file.getParentFile().exists()) {

file.getParentFile().mkdirs();

i++;

System.out.println(i);

}

Writer out = new FileWriter(file);

out.write(answer.text());

out.close();

}

}

try {

Thread.sleep(1000);

} catch (Exception e) {

System.exit(0);// 退出程序

}

}

public void spiderCsdn() throws Exception {

Document document = Jsoup.connect("https://www.csdn.net/")

// 模拟火狐浏览器

// .userAgent("Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)")

.userAgent("Mozilla").get();

Elements url = document.select("#feedlist\_id").select("li").select("div").select("div.title").select("h2")

.select("a");

for (Element question : url) {

String URL = question.attr("abs:href");

Document document2 = Jsoup.connect(URL).userAgent("Mozilla").get();

Elements title = document2.select("#mainBox").select("main").select("div.blog-content-box").select("div")

.select("div").select("div.article-title-box").select("h1");

Elements answer = document2.select("#content\_views").select("p");

System.out.println("\n" + "链接：" + URL + "\n" + "标题：" + title.text() + "\n" + "回答：" + answer.text());

String filepath = "E:" + File.separator + "jsoup" + File.separator + "word" + File.separator + title.text()

+ ".txt";

System.out.println(filepath);

File file = new File(filepath);

if (!file.getParentFile().exists()) {

file.getParentFile().mkdirs();

i++;

System.out.println(i);

}

Writer out = new FileWriter(file);

out.write(answer.text());

out.close();

}

try {

Thread.sleep(1000);

} catch (Exception e) {

System.exit(0);// 退出程序

}

}

}

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.Writer;

import java.util.HashSet;

import java.util.Set;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import org.apdplat.word.WordSegmenter;

public class Filter {

static int chineseNum = 561;

static int engliashNum = 539;

static Set<String> stopWords = new HashSet<>();

public static void main(String[] arg) {

dealChinese();

dealEnglish();

}

static String specialPattern(String string) {

String regEx = "[`~!@#$%^&\*()+=|{}':;',\\[\\].<>/?~！@#￥%……&\*（）——+|{}【】‘；：”“’。，、？]";

Pattern p = Pattern.compile(regEx);

Matcher m = p.matcher(string);

string = m.replaceAll(" ").trim();

string = string.replaceAll("\\s{1,}", " ");

return string;

}

static String chineseWord(String str) {

String reg = "[^\u4e00-\u9fa5]";

str = str.replaceAll(reg, " ");

str = str.replaceAll("\\s{1,}", " ");

return str;

}

static String englishWord(String str) {

String reg = "[^a-zA-Z]";

str = str.replaceAll(reg, " ");

str = str.replaceAll("\\s{1,}", " ");

return str;

}

static void englishStop() {

File file1 = new File("E:" + File.separator + "jsoup" + File.separator + "stopWords.txt");// 停用词

String string1 = null;

BufferedReader br1;

try {

br1 = new BufferedReader(new FileReader(file1));

while ((string1 = br1.readLine()) != null) {// 使用readLine方法，一次读一行 读取停用词

stopWords.add(string1);

}

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} // 构造一个BufferedReader类来读取totalstop文件

catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

static void dealChinese() {

for (int i = 1; i <= chineseNum; i++) {

String str = "";

File in\_file = new File("E:" + File.separator + "jsoup" + File.separator + "News\_1\_Org\_C" + File.separator

+ "News\_1\_Org\_C (" + String.valueOf(i) + ").txt");

try {

FileInputStream in = new FileInputStream(in\_file);

// size 为字串的长度 ，这里一次性读完

int size = in.available();

byte[] buffer = new byte[size];

in.read(buffer);

in.close();

str = new String(buffer, "GBK");

} catch (IOException e) {

e.printStackTrace();

}

str = specialPattern(str);

str = chineseWord(str);

File out\_file = new File("E:" + File.separator + "jsoup" + File.separator + "News\_1\_C" + File.separator

+ "News\_1\_C (" + String.valueOf(i) + ").txt");

if (!out\_file.getParentFile().exists()) {

out\_file.getParentFile().mkdirs();

}

Writer out;

try {

out = new FileWriter(out\_file);

out.write(str);

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

String input = "E:" + File.separator + "jsoup" + File.separator + "News\_1\_C" + File.separator + "News\_1\_C ("

+ String.valueOf(i) + ").txt";

String output = "E:" + File.separator + "jsoup" + File.separator + "News\_1\_Res\_C" + File.separator

+ "News\_1\_Res\_C (" + String.valueOf(i) + ").txt";

try {

WordSegmenter.seg(new File(input), new File(output));

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

static void dealEnglish() {

englishStop();

char[] w = new char[501];

Stemmer s = new Stemmer();

for (int i = 1; i <= engliashNum; i++) {

String str = "";

File in\_file = new File("E:" + File.separator + "jsoup" + File.separator + "News\_1\_Org\_E" + File.separator

+ "News\_1\_Org\_E (" + String.valueOf(i) + ").txt");

try {

FileInputStream in = new FileInputStream(in\_file);

// size 为字串的长度 ，这里一次性读完

int size = in.available();

byte[] buffer = new byte[size];

in.read(buffer);

in.close();

str = new String(buffer, "GBK");

} catch (IOException e) {

e.printStackTrace();

}

str = specialPattern(str);

str = englishWord(str);

File out\_file = new File("E:" + File.separator + "jsoup" + File.separator + "News\_1\_E" + File.separator

+ "News\_1\_E (" + String.valueOf(i) + ").txt");

if (!out\_file.getParentFile().exists()) {

out\_file.getParentFile().mkdirs();

}

Writer out;

try {

out = new FileWriter(out\_file);

out.write(str);

out.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

FileInputStream in = new FileInputStream("E:" + File.separator + "jsoup" + File.separator + "News\_1\_E"

+ File.separator + "News\_1\_E (" + String.valueOf(i) + ").txt");

try {

StringBuilder stringBuilder2 = new StringBuilder();

while (true)

{

int ch = in.read();

if (Character.isLetter((char) ch)) {

int j = 0;

while (true) {

ch = Character.toLowerCase((char) ch);

w[j] = (char) ch;

if (j < 500)

j++;

ch = in.read();

if (!Character.isLetter((char) ch)) {

/\* to test add(char ch) \*/

for (int c = 0; c < j; c++)

s.add(w[c]);

/\* or, to test add(char[] w, int j) \*/

/\* s.add(w, j); \*/

s.stem();

{

// String u;

/\* and now, to test toString() : \*/

if (!stopWords.contains(s.toString())) {

stringBuilder2.append(s.toString());

stringBuilder2.append("\r\n");

}

/\* to test getResultBuffer(), getResultLength() : \*/

/\* u = new String(s.getResultBuffer(), 0, s.getResultLength()); \*/

// System.out.print(u);

}

break;

}

}

File out\_file2 = new File("E:" + File.separator + "jsoup" + File.separator + "News\_1\_Res\_E"

+ File.separator + "News\_1\_Res\_E (" + String.valueOf(i) + ").txt");

if (!out\_file2.getParentFile().exists()) {

out\_file2.getParentFile().mkdirs();

}

Writer out2;

try {

out2 = new FileWriter(out\_file2);

out2.write(stringBuilder2.toString());

out2.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

if (ch < 0)

break;

System.out.print((char) ch);

}

} catch (IOException e) {

System.out.println("error reading ");

break;

}

} catch (Exception e) {

// TODO: handle exception

}

}

}

}

/\*

Porter stemmer in Java. The original paper is in

Porter, 1980, An algorithm for suffix stripping, Program, Vol. 14,

no. 3, pp 130-137,

See also http://www.tartarus.org/~martin/PorterStemmer

History:

Release 1

Bug 1 (reported by Gonzalo Parra 16/10/99) fixed as marked below.

The words 'aed', 'eed', 'oed' leave k at 'a' for step 3, and b[k-1]

is then out outside the bounds of b.

Release 2

Similarly,

Bug 2 (reported by Steve Dyrdahl 22/2/00) fixed as marked below.

'ion' by itself leaves j = -1 in the test for 'ion' in step 5, and

b[j] is then outside the bounds of b.

Release 3

Considerably revised 4/9/00 in the light of many helpful suggestions

from Brian Goetz of Quiotix Corporation (brian@quiotix.com).

Release 4

\*/

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

/\*\*

\* Stemmer, implementing the Porter Stemming Algorithm

\*

\* The Stemmer class transforms a word into its root form. The input word can be

\* provided a character at time (by calling add()), or at once by calling one of

\* the various stem(something) methods.

\*/

class Stemmer {

private char[] b;

private int i, /\* offset into b \*/

i\_end, /\* offset to end of stemmed word \*/

j, k;

private static final int INC = 50;

/\* unit of size whereby b is increased \*/

public Stemmer() {

b = new char[INC];

i = 0;

i\_end = 0;

}

/\*\*

\* Add a character to the word being stemmed. When you are finished adding

\* characters, you can call stem(void) to stem the word.

\*/

public void add(char ch) {

if (i == b.length) {

char[] new\_b = new char[i + INC];

for (int c = 0; c < i; c++)

new\_b[c] = b[c];

b = new\_b;

}

b[i++] = ch;

}

/\*\*

\* Adds wLen characters to the word being stemmed contained in a portion of a

\* char[] array. This is like repeated calls of add(char ch), but faster.

\*/

public void add(char[] w, int wLen) {

if (i + wLen >= b.length) {

char[] new\_b = new char[i + wLen + INC];

for (int c = 0; c < i; c++)

new\_b[c] = b[c];

b = new\_b;

}

for (int c = 0; c < wLen; c++)

b[i++] = w[c];

}

/\*\*

\* After a word has been stemmed, it can be retrieved by toString(), or a

\* reference to the internal buffer can be retrieved by getResultBuffer and

\* getResultLength (which is generally more efficient.)

\*/

public String toString() {

return new String(b, 0, i\_end);

}

/\*\*

\* Returns the length of the word resulting from the stemming process.

\*/

public int getResultLength() {

return i\_end;

}

/\*\*

\* Returns a reference to a character buffer containing the results of the

\* stemming process. You also need to consult getResultLength() to determine the

\* length of the result.

\*/

public char[] getResultBuffer() {

return b;

}

/\* cons(i) is true <=> b[i] is a consonant. \*/

private final boolean cons(int i) {

switch (b[i]) {

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

return false;

case 'y':

return (i == 0) ? true : !cons(i - 1);

default:

return true;

}

}

/\*

\* m() measures the number of consonant sequences between 0 and j. if c is a

\* consonant sequence and v a vowel sequence, and <..> indicates arbitrary

\* presence,

\*

\* <c><v> gives 0 <c>vc<v> gives 1 <c>vcvc<v> gives 2 <c>vcvcvc<v> gives 3 ....

\*/

private final int m() {

int n = 0;

int i = 0;

while (true) {

if (i > j)

return n;

if (!cons(i))

break;

i++;

}

i++;

while (true) {

while (true) {

if (i > j)

return n;

if (cons(i))

break;

i++;

}

i++;

n++;

while (true) {

if (i > j)

return n;

if (!cons(i))

break;

i++;

}

i++;

}

}

/\* vowelinstem() is true <=> 0,...j contains a vowel \*/

private final boolean vowelinstem() {

int i;

for (i = 0; i <= j; i++)

if (!cons(i))

return true;

return false;

}

/\* doublec(j) is true <=> j,(j-1) contain a double consonant. \*/

private final boolean doublec(int j) {

if (j < 1)

return false;

if (b[j] != b[j - 1])

return false;

return cons(j);

}

/\*

\* cvc(i) is true <=> i-2,i-1,i has the form consonant - vowel - consonant and

\* also if the second c is not w,x or y. this is used when trying to restore an

\* e at the end of a short word. e.g.

\*

\* cav(e), lov(e), hop(e), crim(e), but snow, box, tray.

\*

\*/

private final boolean cvc(int i) {

if (i < 2 || !cons(i) || cons(i - 1) || !cons(i - 2))

return false;

{

int ch = b[i];

if (ch == 'w' || ch == 'x' || ch == 'y')

return false;

}

return true;

}

private final boolean ends(String s) {

int l = s.length();

int o = k - l + 1;

if (o < 0)

return false;

for (int i = 0; i < l; i++)

if (b[o + i] != s.charAt(i))

return false;

j = k - l;

return true;

}

/\*

\* setto(s) sets (j+1),...k to the characters in the string s, readjusting k.

\*/

private final void setto(String s) {

int l = s.length();

int o = j + 1;

for (int i = 0; i < l; i++)

b[o + i] = s.charAt(i);

k = j + l;

}

/\* r(s) is used further down. \*/

private final void r(String s) {

if (m() > 0)

setto(s);

}

/\*

\* step1() gets rid of plurals and -ed or -ing. e.g.

\*

\* caresses -> caress ponies -> poni ties -> ti caress -> caress cats -> cat

\*

\* feed -> feed agreed -> agree disabled -> disable

\*

\* matting -> mat mating -> mate meeting -> meet milling -> mill messing -> mess

\*

\* meetings -> meet

\*

\*/

private final void step1() {

if (b[k] == 's') {

if (ends("sses"))

k -= 2;

else if (ends("ies"))

setto("i");

else if (b[k - 1] != 's')

k--;

}

if (ends("eed")) {

if (m() > 0)

k--;

} else if ((ends("ed") || ends("ing")) && vowelinstem()) {

k = j;

if (ends("at"))

setto("ate");

else if (ends("bl"))

setto("ble");

else if (ends("iz"))

setto("ize");

else if (doublec(k)) {

k--;

{

int ch = b[k];

if (ch == 'l' || ch == 's' || ch == 'z')

k++;

}

} else if (m() == 1 && cvc(k))

setto("e");

}

}

/\* step2() turns terminal y to i when there is another vowel in the stem. \*/

private final void step2() {

if (ends("y") && vowelinstem())

b[k] = 'i';

}

/\*

\* step3() maps double suffices to single ones. so -ization ( = -ize plus

\* -ation) maps to -ize etc. note that the string before the suffix must give

\* m() > 0.

\*/

private final void step3() {

if (k == 0)

return;

/\* For Bug 1 \*/ switch (b[k - 1]) {

case 'a':

if (ends("ational")) {

r("ate");

break;

}

if (ends("tional")) {

r("tion");

break;

}

break;

case 'c':

if (ends("enci")) {

r("ence");

break;

}

if (ends("anci")) {

r("ance");

break;

}

break;

case 'e':

if (ends("izer")) {

r("ize");

break;

}

break;

case 'l':

if (ends("bli")) {

r("ble");

break;

}

if (ends("alli")) {

r("al");

break;

}

if (ends("entli")) {

r("ent");

break;

}

if (ends("eli")) {

r("e");

break;

}

if (ends("ousli")) {

r("ous");

break;

}

break;

case 'o':

if (ends("ization")) {

r("ize");

break;

}

if (ends("ation")) {

r("ate");

break;

}

if (ends("ator")) {

r("ate");

break;

}

break;

case 's':

if (ends("alism")) {

r("al");

break;

}

if (ends("iveness")) {

r("ive");

break;

}

if (ends("fulness")) {

r("ful");

break;

}

if (ends("ousness")) {

r("ous");

break;

}

break;

case 't':

if (ends("aliti")) {

r("al");

break;

}

if (ends("iviti")) {

r("ive");

break;

}

if (ends("biliti")) {

r("ble");

break;

}

break;

case 'g':

if (ends("logi")) {

r("log");

break;

}

}

}

/\* step4() deals with -ic-, -full, -ness etc. similar strategy to step3. \*/

private final void step4() {

switch (b[k]) {

case 'e':

if (ends("icate")) {

r("ic");

break;

}

if (ends("ative")) {

r("");

break;

}

if (ends("alize")) {

r("al");

break;

}

break;

case 'i':

if (ends("iciti")) {

r("ic");

break;

}

break;

case 'l':

if (ends("ical")) {

r("ic");

break;

}

if (ends("ful")) {

r("");

break;

}

break;

case 's':

if (ends("ness")) {

r("");

break;

}

break;

}

}

/\* step5() takes off -ant, -ence etc., in context <c>vcvc<v>. \*/

private final void step5() {

if (k == 0)

return;

/\* for Bug 1 \*/ switch (b[k - 1]) {

case 'a':

if (ends("al"))

break;

return;

case 'c':

if (ends("ance"))

break;

if (ends("ence"))

break;

return;

case 'e':

if (ends("er"))

break;

return;

case 'i':

if (ends("ic"))

break;

return;

case 'l':

if (ends("able"))

break;

if (ends("ible"))

break;

return;

case 'n':

if (ends("ant"))

break;

if (ends("ement"))

break;

if (ends("ment"))

break;

/\* element etc. not stripped before the m \*/

if (ends("ent"))

break;

return;

case 'o':

if (ends("ion") && j >= 0 && (b[j] == 's' || b[j] == 't'))

break;

/\* j >= 0 fixes Bug 2 \*/

if (ends("ou"))

break;

return;

/\* takes care of -ous \*/

case 's':

if (ends("ism"))

break;

return;

case 't':

if (ends("ate"))

break;

if (ends("iti"))

break;

return;

case 'u':

if (ends("ous"))

break;

return;

case 'v':

if (ends("ive"))

break;

return;

case 'z':

if (ends("ize"))

break;

return;

default:

return;

}

if (m() > 1)

k = j;

}

/\* step6() removes a final -e if m() > 1. \*/

private final void step6() {

j = k;

if (b[k] == 'e') {

int a = m();

if (a > 1 || a == 1 && !cvc(k - 1))

k--;

}

if (b[k] == 'l' && doublec(k) && m() > 1)

k--;

}

/\*\*

\* Stem the word placed into the Stemmer buffer through calls to add(). Returns

\* true if the stemming process resulted in a word different from the input. You

\* can retrieve the result with getResultLength()/getResultBuffer() or

\* toString().

\*/

public void stem() {

k = i - 1;

if (k > 1) {

step1();

step2();

step3();

step4();

step5();

step6();

}

i\_end = k + 1;

i = 0;

}

/\*\*

\* Test program for demonstrating the Stemmer. It reads text from a a list of

\* files, stems each word, and writes the result to standard output. Note that

\* the word stemmed is expected to be in lower case: forcing lower case must be

\* done outside the Stemmer class. Usage: Stemmer file-name file-name ...

\*/

public static void main(String[] args) {

char[] w = new char[501];

Stemmer s = new Stemmer();

for (int i = 0; i < args.length; i++)

try {

FileInputStream in = new FileInputStream(args[i]);

try {

while (true)

{

int ch = in.read();

if (Character.isLetter((char) ch)) {

int j = 0;

while (true) {

ch = Character.toLowerCase((char) ch);

w[j] = (char) ch;

if (j < 500)

j++;

ch = in.read();

if (!Character.isLetter((char) ch)) {

/\* to test add(char ch) \*/

for (int c = 0; c < j; c++)

s.add(w[c]);

/\* or, to test add(char[] w, int j) \*/

/\* s.add(w, j); \*/

s.stem();

{

String u;

/\* and now, to test toString() : \*/

u = s.toString();

/\* to test getResultBuffer(), getResultLength() : \*/

/\* u = new String(s.getResultBuffer(), 0, s.getResultLength()); \*/

System.out.print(u);

}

break;

}

}

}

if (ch < 0)

break;

System.out.print((char) ch);

}

} catch (IOException e) {

System.out.println("error reading " + args[i]);

break;

}

} catch (FileNotFoundException e) {

System.out.println("file " + args[i] + " not found");

break;

}

}

}