**【信息领域热词分析系统】**

【V1.0】

**源**

**程**

**序**

石家庄铁道大学

目录

[1. 搜索热词解释界面代码](#_Toc14171_WPSOffice_Level1) [3](#_Toc14171_WPSOffice_Level1)

[2. 分类别热词解释界面源代码](#_Toc27165_WPSOffice_Level1) [9](#_Toc27165_WPSOffice_Level1)

[3. 下载功能源代码](#_Toc31043_WPSOffice_Level1) [13](#_Toc31043_WPSOffice_Level1)

[4. 联系图界面源代码](#_Toc14980_WPSOffice_Level1) [17](#_Toc14980_WPSOffice_Level1)

[5. 词云界面源代码](#_Toc7318_WPSOffice_Level1) [24](#_Toc7318_WPSOffice_Level1)

[6. 连接数据库源代码](#_Toc28434_WPSOffice_Level1) [30](#_Toc28434_WPSOffice_Level1)

[7. 操作数据库源代码](#_Toc3833_WPSOffice_Level1) [32](#_Toc3833_WPSOffice_Level1)

[8. 系统类源代码](#_Toc9441_WPSOffice_Level1) [41](#_Toc9441_WPSOffice_Level1)

[9. 爬取文章标题源代码](#_Toc3902_WPSOffice_Level1) [44](#_Toc3902_WPSOffice_Level1)

[10. 爬取热词解释源代码](#_Toc25883_WPSOffice_Level1) [49](#_Toc25883_WPSOffice_Level1)

[11. 文章标题切词源代码](#_Toc6429_WPSOffice_Level1) [51](#_Toc6429_WPSOffice_Level1)

[12. 热词清洗源代码](#_Toc16123_WPSOffice_Level1) [52](#_Toc16123_WPSOffice_Level1)

[13. 统计热词源代码](#_Toc8097_WPSOffice_Level1) [53](#_Toc8097_WPSOffice_Level1)

1. **搜索热词解释界面代码**

<%@ page import=*"wl.\*"*%>

<%@ page import=*"com.model.Key"*%>

<%@ page import=*"com.dao.KeyDao"*%>

<%@ page import=*"java.util.List"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=utf-8"*

pageEncoding=*"utf-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>热词分析</title>

<link rel=*"stylesheet"* href=*"css/linearicons.css"*>

<link rel=*"stylesheet"* href=*"css/font-awesome.min.css"*>

<link rel=*"stylesheet"* href=*"css/bootstrap.css"*>

<link rel=*"stylesheet"* href=*"css/magnific-popup.css"*>

<link rel=*"stylesheet"* href=*"css/nice-select.css"*>

<link rel=*"stylesheet"* href=*"css/animate.min.css"*>

<link rel=*"stylesheet"* href=*"css/owl.carousel.css"*>

<link rel=*"stylesheet"* href=*"css/main.css"*>

<script type=*"text/javascript"*>

function bj() {

alert("下载成功！");

}

</script>

</head>

<body>

<%

String word = request.getParameter("wordd");

String mean3;

String jieshi = request.getParameter("jieshi");

String leibie = request.getParameter("leibie");

String type = "";

System.out.println("aa:"+word);

if (word == null) {

word = "";

mean3 = "";

} else {

if (jieshi != null && !jieshi.equals("")) {

dao.addMean(word, jieshi);

}

if (leibie != null && !leibie.equals("")) {

dao.addleibe(word, leibie);

}

select s = new select();

mean m1 = dao.find\_mean2(word);

mean3 = m1.getMeann();

type = m1.getType();

if (mean3.equals("")) {

s.shuru(word);

mean mean1 = s.mean2;

mean3 = mean1.getMeann();

if(!mean3.equals("该词没有解释")){//该词爬取到了意思

dao.addMean(word, mean3);

}

}

}

%>

<!-- Start Header Area -->

<header id=*"header"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-between d-flex"*>

<div id=*"logo"*>

<a href=*""*><img src=*"img/logo.png"* alt=*""* title=*""* /></a>

<div style="position: *absolute*; right: *3px*; top: *20px*">

<%

KeyDao keyDaoI = new KeyDao();

List<String> typeIndex = null;

typeIndex = keyDaoI.loadKeyType();

int ni = typeIndex.size();

String[] ti = new String[ni];

typeIndex.toArray(ti);

for (int i = 0; i < ni; i++) {

%>

<a href=*"chengXuRenSheng.jsp?word=*<%=ti[i]%>*"* style="color: *white*"><%=ti[i]%></a>&nbsp;&nbsp;&nbsp;

<%

}

%>

<a href=*"Download.jsp"* style="color: *#f77*" onclick="bj()">下载</a> <a

href=*"Connection.jsp"* style="color: *#f77*">联系图</a> <a

href=*"temp.jsp"* style="color: *#f77*">词云</a>

</div>

</div>

</div>

</div>

</header>

<!-- End Header Area -->

<!-- End Header Area -->

<!-- Start Banner Area -->

<section class=*"home-banner-area relative"*>

<div class=*"container"*>

<div

class=*"row fullscreen d-flex align-items-center justify-content-center"*>

<div class=*"banner-content col-lg-8 col-md-12"*>

<h1 class=*"wow fadeIn"* data-wow-duration=*"4s"*>

信息领域 <br>热词分析

</h1>

<div class=*"input-wrap"*>

<form action=*"index.jsp"* method=*"get"*

class=*"form-box d-flex justify-content-between"*>

<input type=*"text"* placeholder=*"请输入搜索内容"* class=*"form-control"*

name=*"wordd"*>

<button type=*"submit"* class=*"btn search-btn"*>Search</button>

<%

select s = new select();

mean mean1 = s.mean2;

mean1.setMeann("该词没有解释");

mean1.setWord("");

%>

</form>

</div>

<h4 class=*"text-white"*>热词</h4>

<div class=*"courses pt-20"*>

<%

typeIndex = keyDaoI.loadKeyType();

typeIndex.toArray(ti);

for (int i = 0; i < ni / 2; i++) {

%>

<a href=*"chengXuRenSheng.jsp?word=*<%=ti[i]%>*"*

data-wow-duration=*"1s"* data-wow-delay=*".9s"*

class=*"primary-btn transparent mr-10 mb-10 wow fadeInDown"*><%=ti[i]%></a>

<%

}

for (int i = ni / 2; i < ni; i++) {

%>

<a href=*"chengXuRenSheng.jsp?word=*<%=ti[i]%>*"*

data-wow-duration=*"1s"* data-wow-delay=*"1.8s"*

class=*"primary-btn transparent mr-10 mb-10 wow fadeInDown"*><%=ti[i]%></a>

<%

}

%>

</div>

</div>

</div>

<div class=*"rocket-img"*>

<img src=*"img/rocket.png"* alt=*""*>

</div>

</section>

<!-- End Banner Area -->

<%

if (word!=null&&!word.equals("")) {

%>

<!-- Start About Area -->

<section class=*"about-area section-gap"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-center"*>

<div class=*"col-lg-5 col-md-6 about-left"*>

<img class=*"img-fluid"* src=*"img/about.jpg"* alt=*""*>

</div>

<div class=*"offset-lg-1 col-lg-6 offset-md-0 col-md-12 about-right"*>

<h1>

<%=word%>

</h1>

<div class=*"wow fadeIn"* data-wow-duration=*"1s"*>

<%

if (type.equals("") && mean3.equals("该词没有解释")) {//没有爬取到意思，数据库中也没有分类

%>

<p>

<%=mean3%><br /> <br />

<form action=*"index.jsp"* method=*"get"*>

<textarea name=*"jieshi"* rows=*"7"* cols=*"70"* value=*""*

placeholder=*"请输入热词解释"*></textarea>

<!-- 输入解释 -->

<br /> <br /> <select name=*"leibie"*>

<!-- 输入类别 -->

<option value=*""*>选择热词类别</option>

<%

typeIndex.toArray(ti);

for (int i = 0; i < ni; i++) {

%>

<option value=*"*<%=ti[i]%>*"*><%=ti[i]%></option>

<%

}

%>

</select> <input type=*"hidden"* name=*"wordd"* value=*"*<%=word%>*"*>

<!-- 点击按钮，依旧查询这个词 -->

<input type=*"submit"* value=*"提交"*

style="position: *absolute*; right: *30px*; width: *100px*">

<!-- 按钮 -->

</form>

</p>

<%

} else if (!type.equals("") && !mean3.equals("该词没有解释")) {//数据库中既有解释也有类别

%>

<p>

<%=mean3%><br /> <br /> 类别：<%=type%>

</p>

<%

}

else if(type.equals("") && !mean3.equals("该词没有解释")){//爬取到了词语得意思，但是数据库中没有分类

%>

<p>

<%=mean3%><br /> <br />

<form action=*"index.jsp"* method=*"get"*>

<select name=*"leibie"*>

<!-- 输入类别 -->

<option value=*""*>选择热词类别</option>

<%

typeIndex.toArray(ti);

for (int i = 0; i < ni; i++) {

%>

<option value=*"*<%=ti[i]%>*"*><%=ti[i]%></option>

<%

}

%>

</select> <input type=*"hidden"* name=*"wordd"* value=*"*<%=word%>*"*>

<!-- 点击按钮，依旧查询这个词 -->

<input type=*"submit"* value=*"提交"*

style="position: *absolute*; right: *30px*; width: *100px*;">

<!-- 按钮 -->

</form>

</p>

<%} %>

</div>

</div>

</div>

</div>

</section>

<!-- End About Area -->

<%

}

%>

<!-- Start Footer Area -->

<footer class=*"footer-area section-gap"*>

</footer>

<!-- End Footer Area -->

<div id=*"back-top"*>

<a title=*"Go to Top"* href=*"#"*></a>

</div>

<!-- ####################### End Scroll to Top Area ####################### -->

<script src=*"js/vendor/jquery-2.2.4.min.js"*></script>

<script src=*"js/vendor/bootstrap.min.js"*></script>

<script src=*"js/easing.min.js"*></script>

<script src=*"js/hoverIntent.js"*></script>

<script src=*"js/superfish.min.js"*></script>

<script src=*"js/jquery.ajaxchimp.min.js"*></script>

<script src=*"js/jquery.magnific-popup.min.js"*></script>

<script src=*"js/owl.carousel.min.js"*></script>

<script src=*"js/owl-carousel-thumb.min.js"*></script>

<script src=*"js/jquery.sticky.js"*></script>

<script src=*"js/jquery.nice-select.min.js"*></script>

<script src=*"js/parallax.min.js"*></script>

<script src=*"js/waypoints.min.js"*></script>

<script src=*"js/wow.min.js"*></script>

<script src=*"js/jquery.counterup.min.js"*></script>

<script src=*"js/mail-script.js"*></script>

<script src=*"js/main.js"*></script>

</body>

</html>

1. 分类别热词解释界面源代码

<%@ page import=*"wl.\*"*%>

<%@ page import=*"java.util.List"*%>

<%@ page import=*"com.model.Key"* %>

<%@ page import=*"com.dao.KeyDao"* %>

<%@ page import=*"java.util.List"* %>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<head>

<title>程序人生</title>

<link rel=*"stylesheet"* href=*"css/linearicons.css"*>

<link rel=*"stylesheet"* href=*"css/font-awesome.min.css"*>

<link rel=*"stylesheet"* href=*"css/magnific-popup.css"*>

<link rel=*"stylesheet"* href=*"css/nice-select.css"*>

<link rel=*"stylesheet"* href=*"css/animate.min.css"*>

<link rel=*"stylesheet"* href=*"css/owl.carousel.css"*>

<link rel=*"stylesheet"* href=*"css/main.css"*>

</head>

<body>

<%

String word=request.getParameter("word");

String mm=dao.find\_mean(word);

if(mm==null||"".equals(mm))

{

select s = new select();

s.shuru(word);

mean mean1 = s.mean2;

mm = mean1.getMeann();

if(mm.equals("该词没有解释")){

}

else{

dao.addMean(word,mm);

}

}

%>

<!-- Start Header Area -->

<header id=*"header"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-between d-flex"*>

<div id=*"logo"*>

<a href=*"index.jsp"*><img src=*"img/logo.png"* alt=*""* title=*""* /></a>

<div style="float:*right*;margin-right:*3px*">

<a href=*"index.jsp"* style="color:*white*" >搜索</a>&nbsp;&nbsp;&nbsp;

<%

KeyDao keyDaoI=new KeyDao();

List<String> typeIndex=null;

typeIndex=keyDaoI.loadKeyType();

int ni=typeIndex.size();

String []ti=new String[ni];

typeIndex.toArray(ti);

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

{

%>

<a href=*"chengXuRenSheng.jsp?word=*<%= ti[i]%>*"* style="color:*white*" ><%= ti[i]%></a>&nbsp;&nbsp;&nbsp;

<%} %>

<a href=*"Connection.jsp"* style="color:*#f77*" >联系图</a>

<a href=*"temp.jsp"* style="color:*#f77*">词云</a>

</div>

</div>

</div>

</div>

</header>

<!-- End Header Area -->

<!-- Start Banner Area -->

<section class=*"banner-area relative"* >

<div class=*"container"*>

<div class=*"row align-items-center justify-content-between d-flex"*>

<div id=*"logo"*>

<a href=*"index.jsp"*><img src=*"img/logo.png"* alt=*""* title=*""* /></a>

</div>

</div>

</div>

<div class=*"container"*>

<div class=*"row d-flex align-items-center justify-content-center"*>

<div class=*"about-content col-lg-12"* style="margin-top:*1.5cm*">

<h1 class=*"text-white"*>

<%=word %>

</h1>

<p><%=mm %></p>

<div class=*"link-nav"*>

<span class=*"box"*>

<a href=*"index.jsp"*>Home </a>

</span>

</div>

</div>

</div>

</div>

<div class=*"rocket-img"* >

<img src=*"img/rocket.png"* alt=*""*>

</div>

</section>

<!-- End Banner Area -->

<!-- Start Faculty Area -->

<section class=*"faculty-area section-gap"* style="background-color:*#bef*">

<br/>

<br/>

<%

List<mean> means=dao.find\_allwords1(word);

for(mean m:means){

%><div class=*"content"* width=*"70%"*>

<div class=*"row justify-content-center d-flex align-items-center"*>

<div class=*"col-lg-3 col-md-6 col-sm-12 single-faculty"*>

<div class=*"meta-text text-center"*>

<h4><%=m.getWord() %></h4>

<div class=*"info wow fadeIn"* data-wow-duration=*"1s"* data-wow-delay=*".1s"*>

<p>

<%=m.getMeann()%>

</p>

</div>

<div >

<%

List<url> urls=dao.find\_allUrls2(m.getWord() );

for(url u:urls){

%>

<a href=*"*<%=u.getUrll() %>*"*><%=u.getTitle() %></a><br/>

<%} %>

</div>

</div>

</div>

</div>

</div>

<br/>

<br/>

<br/>

<%} %>

</section>

<!-- End Faculty Area -->

<!-- Start Footer Area -->

<footer class=*"footer-area section-gap"*>

</footer>

<!-- End Footer Area -->

<div id=*"back-top"*>

<a title=*"Go to Top"* href=*"#"*></a>

</div>

<!-- ####################### End Scroll to Top Area ####################### -->

<script src=*"js/vendor/jquery-2.2.4.min.js"*></script>

<script src=*"js/vendor/bootstrap.min.js"*></script>

<script src=*"js/easing.min.js"*></script>

<script src=*"js/hoverIntent.js"*></script>

<script src=*"js/superfish.min.js"*></script>

<script src=*"js/jquery.ajaxchimp.min.js"*></script>

<script src=*"js/jquery.magnific-popup.min.js"*></script>

<script src=*"js/owl.carousel.min.js"*></script>

<script src=*"js/owl-carousel-thumb.min.js"*></script>

<script src=*"js/jquery.sticky.js"*></script>

<script src=*"js/jquery.nice-select.min.js"*></script>

<script src=*"js/parallax.min.js"*></script>

<script src=*"js/waypoints.min.js"*></script>

<script src=*"js/wow.min.js"*></script>

<script src=*"js/jquery.counterup.min.js"*></script>

<script src=*"js/mail-script.js"*></script>

<script src=*"js/main.js"*></script>

</body>

</html>

1. **下载功能源代码**

package wl;

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public class File {

static int count=1;

static int c=0;

static String[] q= {"一","二","三","四","五","六","七"};

public static void WriteStringToFile2(List<mean> means,String type,String name) {

try {

FileWriter fw = new FileWriter(name, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.append(q[c]+"."+type);//类别

bw.write("\n");

for(mean model:means) {

String word=model.getWord();//标题

bw.write(count+"."+model.getWord());

bw.write("\n");

bw.write(" ");

bw.write(model.getMeann());

bw.write("\r\n");

count++;

}

bw.close();

fw.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void down() {

c=0;

String[] type= {"程序人生","大数据","运维","编程语言","区块链","架构","人工智能"};

Date day=new Date();

SimpleDateFormat df = new SimpleDateFormat("yyyyMMddHHmmss");

String name="C:\\Users\\lenovo\\Desktop\\热词解释"+df.format(day)+".docx";

mulu(name);

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

count=1;

List<mean> words=dao.find\_allwords1(type[i]);

WriteStringToFile2(words,type[i],name);

c++;

}

}

public static void mulu(String name) {

try {

FileWriter fw = new FileWriter(name, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.append(" 信息领域热词解释");//类别

bw.write("\n");

bw.write("\n");

bw.write("\n");

bw.write("一.程序人生....................................................................1");

bw.write("\n");

bw.write("二.大数据......................................................................2");

bw.write("\n");

bw.write("三.运维........................................................................2");

bw.write("\n");

bw.write("四.编程语言....................................................................3");

bw.write("\n");

bw.write("五.区块链......................................................................4");

bw.write("\n");

bw.write("六.架构....................................................................... 4");

bw.write("\n");

bw.write("六.人工智能....................................................................5");

bw.write("\n");

bw.write("\n");

bw.write("\n");

bw.write("\n");

bw.write("\n");

bw.write("\n");

bw.close();

fw.close();

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

package com.Servlet;

import com.alibaba.fastjson.JSONObject;

import com.dao.KeyDao;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class wordServlet

\* @WebServlet("/wordServlet")

\*/

public class wordServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public wordServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

doPost(request,response);

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

request.setCharacterEncoding("utf-8");//定义content的字符格式

String content = request.getParameter("content");//定义content

//System.out.println("获取到的内容:"+content);

JSONObject json = new JSONObject();//定义一个json

KeyDao wordDao = new KeyDao();//调用方法

String explain = wordDao.loadKeyCon(content);//

json.put("explain",explain);//相当于定义一个json的属性，属性值是爬取出来的explain

response.setCharacterEncoding("utf-8");//字符串的格式

response.getWriter().write(json.toJSONString());//往回传

}

}

1. **联系图界面源代码**

<%@ page import=*"wl.\*"*%>

<%@ page import=*"com.model.Key"*%>

<%@ page import=*"com.dao.KeyDao"*%>

<%@ page import=*"java.util.List"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=utf-8"*

pageEncoding=*"utf-8"*%>

<%@ page import=*"com.model.Key"*%>

<%@ page import=*"com.dao.KeyDao"*%>

<%@ page import=*"java.util.List"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=utf-8"*>

<title>联系图</title>

<script src=*"js/echarts.js"*></script>

<link rel=*"stylesheet"* href=*"css/linearicons.css"*>

<link rel=*"stylesheet"* href=*"css/font-awesome.min.css"*>

<link rel=*"stylesheet"* href=*"css/bootstrap.css"*>

<link rel=*"stylesheet"* href=*"css/magnific-popup.css"*>

<link rel=*"stylesheet"* href=*"css/nice-select.css"*>

<link rel=*"stylesheet"* href=*"css/animate.min.css"*>

<link rel=*"stylesheet"* href=*"css/owl.carousel.css"*>

<link rel=*"stylesheet"* href=*"css/main.css"*>

<script type=*"text/javascript"*>

function bj(){

alert("下载成功！");

}

</script>

</head>

<body>

<%

String word = request.getParameter("name");

String mean3;

if (word == null) {

word = "";

mean3 = "";

} else {

mean3 = dao.find\_mean(word);

}

%>

<header id=*"header"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-between d-flex"*>

<div id=*"logo"*>

<a href=*""*><img src=*"img/logo.png"* alt=*""* title=*""* /></a>

<div style="position: *absolute*; right: *3px*; top: *20px*">

<a href=*"index.jsp"* style="color:*white*" >搜索</a>&nbsp;&nbsp;&nbsp;

<%

KeyDao keyDaoI=new KeyDao();

List<String> typeIndex=null;

typeIndex=keyDaoI.loadKeyType();

int ni=typeIndex.size();

String []ti=new String[ni];

typeIndex.toArray(ti);

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

{

%>

<a href=*"chengXuRenSheng.jsp?word=*<%= ti[i]%>*"* style="color:*white*" ><%= ti[i]%></a>&nbsp;&nbsp;&nbsp;

<%} %>

<a href=*"Connection.jsp"* style="color:*#f77*" >联系图</a>&nbsp;&nbsp;&nbsp;

<a href=*"temp.jsp"* style="color:*#f77*">词云</a>

</div>

</div>

</div>

</div>

</header>

<section class=*"home-banner-area relative"*>

<div class=*"container"*>

<div

class=*"row fullscreen d-flex align-items-center justify-content-center"*>

<div class=*"banner-content col-lg-8 col-md-12"*>

<div class=*"panel panel-default"*>

<div class=*"panel-body"*

style="width: *auto*; height: *500px*; margin-top: *2cm*">

<div id=*"chart1"* style="width: *100%*; height: *100%*"></div>

</div>

</div>

</div>

</div>

</div>

<div align=*"center"*>

<img src=*"img/rocket.png"* alt=*""*>

</div>

</section>

<!-- End Banner Area -->

<% if(!"".equals(word)&&word!=null&&!"".equals(mean3)&&mean3!=null){%>

<!-- Start About Area -->

<section class=*"about-area section-gap"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-center"*>

<div class=*"col-lg-5 col-md-6 about-left"*>

<img class=*"img-fluid"* src=*"img/about.jpg"* alt=*""*>

</div>

<div class=*"offset-lg-1 col-lg-6 offset-md-0 col-md-12 about-right"*>

<h1>

<%=word %>

</h1>

<div class=*"wow fadeIn"* data-wow-duration=*"1s"*>

<p>

<%=mean3 %>

</p>

</div>

</div>

</div>

</div>

</section>

<!-- End About Area -->

<%} %>

<!-- Start Footer Area -->

<footer class=*"footer-area section-gap"*>

<div class=*"container"*>

<div class=*"col-lg-4 col-md-6 single-footer-widget"*>

<p class=*"footer-text m-0 col-lg-8 col-md-12"*>

Copyright &copy;

<script>

document.write(new Date().getFullYear());

</script>

</p>

</div>

</div>

</footer>

<!-- End Footer Area -->

<div id=*"back-top"*>

<a title=*"Go to Top"* href=*"#"*></a>

</div>

<!-- ####################### End Scroll to Top Area ####################### -->

<script src=*"js/vendor/jquery-2.2.4.min.js"*></script>

<script src=*"js/vendor/bootstrap.min.js"*></script>

<script src=*"js/easing.min.js"*></script>

<script src=*"js/hoverIntent.js"*></script>

<script src=*"js/superfish.min.js"*></script>

<script src=*"js/jquery.ajaxchimp.min.js"*></script>

<script src=*"js/jquery.magnific-popup.min.js"*></script>

<script src=*"js/owl.carousel.min.js"*></script>

<script src=*"js/owl-carousel-thumb.min.js"*></script>

<script src=*"js/jquery.sticky.js"*></script>

<script src=*"js/jquery.nice-select.min.js"*></script>

<script src=*"js/parallax.min.js"*></script>

<script src=*"js/waypoints.min.js"*></script>

<script src=*"js/wow.min.js"*></script>

<script src=*"js/jquery.counterup.min.js"*></script>

<script src=*"js/mail-script.js"*></script>

<script src=*"js/main.js"*></script>

</body>

</html>

<script type=*"text/javascript"*>

var chart1 = echarts.init(document.getElementById("chart1"));

chart1.showLoading();

init();

<%KeyDao keyDaoC = new KeyDao();

List<String> types = null;%>

function init(){

chart1.hideLoading();

var option = {

backgroundColor: '#ffffff', // 背景颜色

title: { // 图表标题

text: "热词关系", // 标题文本

//left : '3%', // 标题距离左侧边距

//top : '3%', // 标题距顶部边距

textStyle : { // 标题样式

color : '#3174c7', // 标题字体颜色

fontSize : '24', // 标题字体大小

}

},

tooltip: { // 提示框的配置

formatter: function(param) {

if (param.dataType === 'edge') {

//return param.data.category + ': ' + param.data.target;

return param.data.target;

}

//return param.data.category + ': ' + param.data.name;

return param.data.name;

}

},

series: [{

type: "graph", // 系列类型:关系图

top: '10%', // 图表距离容器顶部的距离

roam: true, // 是否开启鼠标缩放和平移漫游。默认不开启。如果只想要开启缩放或者平移，可以设置成 'scale' 或者 'move'。设置成 true 为都开启

focusNodeAdjacency: true, // 是否在鼠标移到节点上的时候突出显示节点以及节点的边和邻接节点。[ default: false ]

force: { // 力引导布局相关的配置项，力引导布局是模拟弹簧电荷模型在每两个节点之间添加一个斥力，每条边的两个节点之间添加一个引力，每次迭代节点会在各个斥力和引力的作用下移动位置，多次迭代后节点会静止在一个受力平衡的位置，达到整个模型的能量最小化。

// 力引导布局的结果有良好的对称性和局部聚合性，也比较美观。

repulsion: 80, // [ default: 50 ]节点之间的斥力因子(关系对象之间的距离)。支持设置成数组表达斥力的范围，此时不同大小的值会线性映射到不同的斥力。值越大则斥力越大

edgeLength: [100, 50] // [ default: 30 ]边的两个节点之间的距离(关系对象连接线两端对象的距离,会根据关系对象值得大小来判断距离的大小)，

// 这个距离也会受 repulsion。支持设置成数组表达边长的范围，此时不同大小的值会线性映射到不同的长度。值越小则长度越长。如下示例:

// 值最大的边长度会趋向于 10，值最小的边长度会趋向于 50 edgeLength: [10, 50]

},

layout: "force", // 图的布局。[ default: 'none' ]

// 'none' 不采用任何布局，使用节点中提供的 x， y 作为节点的位置。

// 'circular' 采用环形布局;'force' 采用力引导布局.

// 标记的图形

//symbol: "path://M19.300,3.300 L253.300,3.300 C262.136,3.300 269.300,10.463 269.300,19.300 L269.300,21.300 C269.300,30.137 262.136,37.300 253.300,37.300 L19.300,37.300 C10.463,37.300 3.300,30.137 3.300,21.300 L3.300,19.300 C3.300,10.463 10.463,3.300 19.300,3.300 Z",

symbol: 'circle',

lineStyle: { // 关系边的公用线条样式。其中 lineStyle.color 支持设置为'source'或者'target'特殊值，此时边会自动取源节点或目标节点的颜色作为自己的颜色。

normal: {

color: 'grey', // 线的颜色[ default: '#aaa' ]

width: 1, // 线宽[ default: 1 ]

type: 'solid', // 线的类型[ default: solid ] 'dashed' 'dotted'

opacity: 0.5, // 图形透明度。支持从 0 到 1 的数字，为 0 时不绘制该图形。[ default: 0.5 ]

curveness: 0.2 // 边的曲度，支持从 0 到 1 的值，值越大曲度越大。[ default: 0 ]

}

},

label: { // 关系对象上的标签

normal: {

show: true, // 是否显示标签

position: "inside", // 标签位置:'top''left''right''bottom''inside''insideLeft''insideRight''insideTop''insideBottom''insideTopLeft''insideBottomLeft''insideTopRight''insideBottomRight'

textStyle: { // 文本样式

fontSize: 12

}

}

},

edgeLabel: { // 连接两个关系对象的线上的标签

normal: {

show: true,

textStyle: {

fontSize: 8

},

formatter: function(param) { // 标签内容

return param.data.category;

}

}

},

data: [

<%types = keyDaoC.loadKeyType();

for (String type : types) {

if (!"人工智能".equals(type)) {%>

{

name: "<%=type%>",

draggable: true,// 节点是否可拖拽，只在使用力引导布局的时候有用。

symbolSize: [60, 60],

category: "<%=type%>"

},

<%}

//System.out.println("datas+"+type);

List<Key> keysC = keyDaoC.loadKey(type);

for (Key key : keysC) {%>

{

name: "<%=key.getKey()%>",

draggable: true,// 节点是否可拖拽，只在使用力引导布局的时候有用。

symbolSize: [50, 50],

category: "<%=type%>" // 数据项所在类目的 index。

},

<%}

}%>

],

categories: [

<%types = keyDaoC.loadKeyType();

for (String type : types) {

//System.out.println("cats+"+type);%>

{name:'<%=type%>'},

<%}%>

],

links: [

<%types = keyDaoC.loadKeyType();

int nn = types.size();

String[] ts = new String[nn];

types.toArray(ts);

int ns = nn - 1;

for (int a = 0; a < ns; a++) {%>

{

target:"<%=ts[a]%>",

source: "<%=ts[a + 1]%>",

category:"连接"

},

<%}

for (String type : types) {

//System.out.println("links+" + type);

List<Key> keysC = keyDaoC.loadKey(type);

for (Key key : keysC) {

//System.out.println("links+"+key.getKey());%>

{

target:"<%=key.getKey()%>",

source: "<%=type%>",

category:""

},

<%}

}%>

]

}],

animationEasingUpdate: "quinticInOut", // 数据更新动画的缓动效果。[ default: cubicOut ] "quinticInOut"

animationDurationUpdate: 100 // 数据更新动画的时长。[ default: 300 ]

};

// 使用刚指定的配置项和数据显示图表

chart1.setOption(option);

}

chart1.on("click",function(param){

var name=param.name;

window.location.href="Connection.jsp?name="+name;

});

</script>

1. **词云界面源代码**

<%@ page import=*"wl.\*"*%>

<%@ page import=*"com.model.Key"*%>

<%@ page import=*"com.dao.KeyDao"*%>

<%@ page import=*"java.util.List"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=utf-8"*

pageEncoding=*"utf-8"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<title>词云</title>

<link rel=*"stylesheet"* href=*"css/linearicons.css"*>

<link rel=*"stylesheet"* href=*"css/font-awesome.min.css"*>

<link rel=*"stylesheet"* href=*"css/bootstrap.css"*>

<link rel=*"stylesheet"* href=*"css/magnific-popup.css"*>

<link rel=*"stylesheet"* href=*"css/nice-select.css"*>

<link rel=*"stylesheet"* href=*"css/animate.min.css"*>

<link rel=*"stylesheet"* href=*"css/owl.carousel.css"*>

<link rel=*"stylesheet"* href=*"css/main.css"*>

<script type=*"text/javascript"*>

function bj(){

alert("下载成功！");

}

</script>

<script src=*"cloudJs/echarts.min.js"*></script>

<script src=*"cloudJs/echarts-wordcloud.min.js"*></script>

</head>

<body>

<%

String word = request.getParameter("name");

String mean3;

if (word == null) {

word = "";

mean3 = "";

} else {

mean3 =dao.find\_mean(word);

}

%>

<!-- Start Header Area -->

<header id=*"header"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-between d-flex"*>

<div id=*"logo"*>

<a href=*""*><img src=*"img/logo.png"* alt=*""* title=*""* /></a>

<div style="position:*absolute*;right:*3px*;top:*20px*">

<a href=*"index.jsp"* style="color:*white*" >搜索</a>&nbsp;&nbsp;&nbsp;

<%

KeyDao keyDaoI=new KeyDao();

List<String> typeIndex=null;

typeIndex=keyDaoI.loadKeyType();

int ni=typeIndex.size();

String []ti=new String[ni];

typeIndex.toArray(ti);

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

{

%>

<a href=*"chengXuRenSheng.jsp?word=*<%= ti[i]%>*"* style="color:*white*" ><%= ti[i]%></a>&nbsp;&nbsp;&nbsp;

<%} %>

<a href=*"Connection.jsp"* style="color:*#f77*" >联系图</a>

<a href=*"temp.jsp"* style="color:*#f77*">词云</a>

</div>

</div>

</div>

</div>

</header>

<!-- End Header Area -->

<!-- Start Banner Area -->

<section class=*"home-banner-area relative"*>

<div class=*"container"*>

<div

class=*"row fullscreen d-flex align-items-center justify-content-center"*>

<div class=*"banner-content col-lg-8 col-md-12"*>

<div class=*"panel panel-default"*>

<div style="width: *450px*; height: *450px*;background-color:*white*;margin-left:*100px*;margin-top:*1.5cm*;" >

<div id=*"worldCloud"* style="width: *100%*; height: *100%*"></div>

</div>

</div>

</div>

</div>

</div>

<div align=*"center"*>

<img src=*"img/rocket.png"* alt=*""*>

</div>

</section>

<!-- End Banner Area -->

<% if(!"".equals(word)&&word!=null&&!"".equals(mean3)&&mean3!=null){%>

<!-- Start About Area -->

<section class=*"about-area section-gap"*>

<div class=*"container"*>

<div class=*"row align-items-center justify-content-center"*>

<div class=*"col-lg-5 col-md-6 about-left"*>

<img class=*"img-fluid"* src=*"img/about.jpg"* alt=*""*>

</div>

<div class=*"offset-lg-1 col-lg-6 offset-md-0 col-md-12 about-right"*>

<h1>

<%=word %>

</h1>

<div class=*"wow fadeIn"* data-wow-duration=*"1s"*>

<p>

<%=mean3 %>

</p>

</div>

</div>

</div>

</div>

</section>

<!-- End About Area -->

<%} %>

<!-- End Courses Area -->

<%

KeyDao keyDao = new KeyDao();

List<Key> keys = keyDao.loadKey();

%>

<script type=*"text/javascript"*>

var worldCloudcharts=echarts.init(document.getElementById('worldCloud'));

var worldCloudoption = {

title: {

text: '热词TOP50',

textStyle: {

fontSize: 50,

color:'#FFFFFF'

}

},

tooltip: {

show: true

},

series: [{

name: '热词TOP50',

type: 'wordCloud',

sizeRange: [6, 66],

rotationRange: [-45, 90],

textPadding: 0,

autoSize: {

enable: true,

minSize: 6

},

textStyle: {

normal: {

color: function() {

return 'rgb(' + [

Math.round(Math.random() \* 160),

Math.round(Math.random() \* 160),

Math.round(Math.random() \* 160)

].join(',') + ')';

}

},

emphasis: {

shadowBlur: 10,

shadowColor: '#333'

}

},

data: [

<%int i = 0;

for (Key key : keys) {%>

{

name: "<%=key.getKey()%>",

value: "<%=key.getNum()%>"

<%i++;%>

}

<%if (i != keys.size()) {%>

,

<%}

}%>

]

}]

};

var JosnList = [];

JosnList.push(

<%i = 0;

for (Key key : keys) {%>

{

name: "<%=key.getKey()%>",

value: "<%=key.getNum()%>"

<%i++;%>

}

<%if (i != keys.size()) {%>

,

<%}

}%>

);

worldCloudoption.series[0].data = JosnList;

worldCloudcharts.setOption(worldCloudoption);

//添加点击事件（单击），还有其他鼠标事件和键盘事件等等

worldCloudcharts.on("click",function(param){

var name=param.name;

window.location.href="temp.jsp?name="+name;

});

</script>

<!--点击按钮-->

<!-- Start Footer Area -->

<footer class=*"footer-area section-gap"*>

<div class=*"container"*>

<div class=*"col-lg-4 col-md-6 single-footer-widget"*>

<p class=*"footer-text m-0 col-lg-8 col-md-12"*>

Copyright &copy;

<script>

document.write(new Date().getFullYear());

</script>

</p>

</div>

</div>

</footer>

<!-- End Footer Area -->

<div id=*"back-top"*>

<a title=*"Go to Top"* href=*"#"*></a>

</div>

<!-- ####################### End Scroll to Top Area ####################### -->

<script src=*"js/vendor/jquery-2.2.4.min.js"*></script>

<script src=*"js/vendor/bootstrap.min.js"*></script>

<script src=*"js/easing.min.js"*></script>

<script src=*"js/hoverIntent.js"*></script>

<script src=*"js/superfish.min.js"*></script>

<script src=*"js/jquery.ajaxchimp.min.js"*></script>

<script src=*"js/jquery.magnific-popup.min.js"*></script>

<script src=*"js/owl.carousel.min.js"*></script>

<script src=*"js/owl-carousel-thumb.min.js"*></script>

<script src=*"js/jquery.sticky.js"*></script>

<script src=*"js/jquery.nice-select.min.js"*></script>

<script src=*"js/parallax.min.js"*></script>

<script src=*"js/waypoints.min.js"*></script>

<script src=*"js/wow.min.js"*></script>

<script src=*"js/jquery.counterup.min.js"*></script>

<script src=*"js/mail-script.js"*></script>

<script src=*"js/main.js"*></script>

</body>

</html>

1. **连接数据库源代码**

package com.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class DBUtil {

public static Connection getConnection() {

try {

//1建立连接

Class.forName("oracle.jdbc.driver.OracleDriver").newInstance();

//System.out.println("数据库驱动加载成功");

} catch (InstantiationException | IllegalAccessException | ClassNotFoundException e) {

// TODO Auto-generated catch block

System.out.println("数据库驱动加载失败");

e.printStackTrace();

}

String user = "moon";//登录oracle的登录名

String password = "root";

//url jdbc:oracle:thin表示jdbc连接oracle数据库的通讯协议

//@localhost主机

//1521主机上的程序端口

//orcl程序端口上的数据库

String url = "jdbc:oracle:thin:@localhost:1521:orcl";

Connection connection = null;

try {

connection = DriverManager.getConnection(url,user,password);

} catch (SQLException e) {

System.out.println("连接失败");

e.printStackTrace();

}

return connection;

}

public static void close(Connection connection ) {

try {

if (connection != null) {

connection.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void close(PreparedStatement preparedStatement ) {

try {

if (preparedStatement != null) {

preparedStatement.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static void close(ResultSet resultSet ) {

try {

if (resultSet != null) {

resultSet.close();

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

1. **操作数据库源代码**

package com.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import com.model.Key;

import com.util.DBUtil;

import com.util.UserException;

public class KeyDao implements IKey{

@Override

public List<String> loadKeyContent() {

//select \* FROM ( SELECT key from sort ORDER BY NUM DESC)WHERE ROWNUM < 21;

// TODO Auto-generated method stub

Connection connection=DBUtil.getConnection();

String sql="select \* FROM ( SELECT key from MOON.sort ORDER BY NUM DESC)WHERE ROWNUM < 50";

PreparedStatement preparedStatement=null;

ResultSet resultSet=null;

//String[] keys=new []String;

List <String>keys=new ArrayList<String>();

String key=null;

try {

System.out.println(sql);

preparedStatement=connection.prepareStatement(sql);

resultSet=preparedStatement.executeQuery();

while(resultSet.next())

{

key=new String();

key=resultSet.getString("key");

System.out.println(key);

keys.add(key);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return keys;

}

@Override

public List<Key> loadKey(String type) {

Connection connection=DBUtil.getConnection();

String sql=null;

if(!"".equals(type)&&type!=null)

{

sql="SELECT key from sort where type='"+type+"'";

}

PreparedStatement preparedStatement=null;

ResultSet resultSet=null;

//String[] keys=new []String;

List <Key>keys=new ArrayList<Key>();

Key key;

try {

preparedStatement=connection.prepareStatement(sql);

resultSet=preparedStatement.executeQuery();

while(resultSet.next())

{

key=new Key();

key.setKey(resultSet.getString("key"));

keys.add(key);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return keys;

}

@Override

public List<Key> loadKey() {

Connection connection=DBUtil.getConnection();

String sql="select \* FROM ( SELECT key,num from MOON.sort ORDER BY NUM DESC)WHERE ROWNUM < 50";

PreparedStatement preparedStatement=null;

ResultSet resultSet=null;

//String[] keys=new []String;

List <Key>keys=new ArrayList<Key>();

Key key;

try {

System.out.println(sql);

preparedStatement=connection.prepareStatement(sql);

resultSet=preparedStatement.executeQuery();

while(resultSet.next())

{

key=new Key();

key.setKey(resultSet.getString("key"));

key.setNum(resultSet.getString("num"));

keys.add(key);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return keys;

}

@Override

public String loadKeyCon(String content) {

Connection connection=DBUtil.getConnection();

String sql = null;

if(!"".equals(content)&&content!=null)

{

sql="SELECT content from detail where key = '"+content+ "'";

}

PreparedStatement preparedStatement=null;

ResultSet resultSet=null;

//String[] keys=new []String;

String result=null;

try {

preparedStatement=connection.prepareStatement(sql);

resultSet=preparedStatement.executeQuery();

System.out.println(sql);

while(resultSet.next())

{

result=resultSet.getString("content");

System.out.println(result);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return result;

}

@Override

public List<String> loadKeyType() {

Connection connection=DBUtil.getConnection();

String sql = null;

sql="SELECT DISTINCT type from sort ";

PreparedStatement preparedStatement=null;

ResultSet resultSet=null;

List <String>types=new ArrayList<String>();

String type=null;

try {

preparedStatement=connection.prepareStatement(sql);

resultSet=preparedStatement.executeQuery();

while(resultSet.next())

{

type=resultSet.getString("type");

types.add(type);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return types;

}

}

package wl;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import com.util.DBUtil;

public class dao {

public static List<mean> find\_allwords1(String type) {

// 用户登录

Connection connection = DBUtil.getConnection();

String sql = "select DETAIL.KEY,DETAIL.CONTENT from SORT join DETAIL on SORT.KEY=DETAIL.KEY WHERE TYPE ='"

+ type + "'";

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

List<mean> means = new ArrayList<mean>();

try {

System.out.println("sql:" + sql);

preparedStatement = connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

mean m = new mean();

m.setWord(resultSet.getString("KEY"));

m.setMeann(resultSet.getString("CONTENT"));

means.add(m);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return means;

}

public static String find\_mean(String word) {

// 用户登录

Connection connection = DBUtil.getConnection();

String sql = "select DETAIL.CONTENT from DETAIL WHERE KEY='" + word + "'";

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

String mean = "";

try {

System.out.println("sql:" + sql);

preparedStatement = connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

mean = resultSet.getString("CONTENT");

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return mean;

}

public static mean find\_mean2(String word) {//查询出词的意思以及类别

// 用户登录

Connection connection = DBUtil.getConnection();

String sql = "select DETAIL.CONTENT,SORT.TYPE from DETAIL join SORT on DETAIL.KEY=SORT.KEY WHERE DETAIL.KEY='" + word + "'";

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

mean mean1=new mean();

mean1.setMeann("");

mean1.setType("");

try {

System.out.println("sql:" + sql);

preparedStatement = connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

mean1.setMeann(resultSet.getString("CONTENT"));

mean1.setType(resultSet.getString("TYPE"));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return mean1;

}

public static List<url> find\_allUrls2(String word) {

// 用户登录

Connection connection = DBUtil.getConnection();

String sql = "select DISTINCT URL.KEY,URL.URL,TITLEE.TITLE from URL join TITLEE on URL.URL=TITLEE.URL WHERE URL.KEY ='"

+ word + "'";

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

List<url> means = new ArrayList<url>();

try {

System.out.println("sql:" + sql);

preparedStatement = connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

url m = new url();

m.setWord(resultSet.getString("KEY"));

m.setUrll(resultSet.getString("URL"));

m.setTitle(resultSet.getString("TITLE"));

means.add(m);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return means;

}

public static int find\_countWord(String word) {//查询该词是否有解释

// 用户登录

Connection connection = DBUtil.getConnection();

String sql = "select count(KEY) from DETAIL where KEY ='" + word + "'";

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

int count = 0;

try {

System.out.println("sql:" + sql);

preparedStatement = connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

count = resultSet.getInt("count(KEY)");

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return count;

}

public static int find\_countWord2(String word) {//查询该词是否分类

// 用户登录

Connection connection = DBUtil.getConnection();

String sql = "select count(KEY) from SORT where KEY ='" + word + "'";

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

int count = 0;

try {

System.out.println("sql:" + sql);

preparedStatement = connection.prepareStatement(sql);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

count = resultSet.getInt("count(KEY)");

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

return count;

}

public static void addMean(String word, String mean) {

int count = find\_countWord(word);

System.out.println("count:"+count);

if (count > 0) {

} else {

// 获得连接对象

Connection connection = DBUtil.getConnection();

// 创建语句传输对象

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

try {

String sql = "INSERT into DETAIL(KEY,CONTENT) VALUES ('" + word + "','" + mean + "')";

preparedStatement = connection.prepareStatement(sql);

System.out.println("sql" + sql);

preparedStatement.executeUpdate();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

// 关闭资源

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

}

}

public static void addleibe(String word, String leibe) {

int count = find\_countWord2(word);

System.out.println("count:"+count);

if (count > 0) {

} else {

// 获得连接对象

Connection connection = DBUtil.getConnection();

// 创建语句传输对象

PreparedStatement preparedStatement = null;

ResultSet resultSet = null;

try {

String sql = "INSERT into SORT(KEY,NUM,TYPE) VALUES ('" +word +"','10','"+leibe+"')";

preparedStatement = connection.prepareStatement(sql);

System.out.println("sql" + sql);

preparedStatement.executeUpdate();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

// 关闭资源

DBUtil.close(resultSet);

DBUtil.close(preparedStatement);

DBUtil.close(connection);

}

}

}

}

1. **系统类源代码**

package com.model;

public class Key {

private String key;//热词名字

private String num;//热词出现次数

private String type;//热词种类

private String url;//热词链接

private String content;//热词内容

public Key() {}

public Key(String key,String num,String type,String url,String content)

{

this.key=key;

this.num=num;

this.key=key;

this.url=url;

this.content=content;

}

public String getKey() {

return key;

}

public void setKey(String key) {

this.key = key;

}

public String getNum() {

return num;

}

public void setNum(String num) {

this.num = num;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

}

package wl;

public class mean {

private String word;

private String meann;

private String type;

public String getWord() {

return word;

}

public void setWord(String word) {

this.word = word;

}

public String getMeann() {

return meann;

}

public void setMeann(String meann) {

this.meann = meann;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

}

package wl;

public class url {

private String word;

private String urll;

private String title;

public String getWord() {

return word;

}

public void setWord(String word) {

this.word = word;

}

public String getUrll() {

return urll;

}

public void setUrll(String urll) {

this.urll = urll;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

}

1. **爬取文章标题源代码**

package zuoye1;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import us.codecraft.webmagic.Page;

import us.codecraft.webmagic.Site;

import us.codecraft.webmagic.Spider;

import us.codecraft.webmagic.processor.PageProcessor;

public class BlogPageProcessor implements PageProcessor {

private static Connection conn = null;

private static PreparedStatement ps = null;

// 标题和链接获取

private static String TITLEQUERY = "div.post\_item\_body h3 a.titlelnk";

// 作者

private static String AUTHORQUERY = "div.post\_item\_foot a.lightblue ";

//初始化带爬取网页地址

private static List<String> urls() {

List listUrl=new ArrayList<String>();

for(int i=171;i<=200;i++) {

listUrl.add("http://www.cnblogs.com/"+i);

}

listUrl.toArray(new String[listUrl.size()]);

return listUrl;

}

//jsoup根据html字符串和语法来获取内容

private static String selectDocumentText(String htmlText,String Query) {

Document doc=Jsoup.parse(htmlText);

String select=doc.select(Query).text();

return select;

}

//jsoup根据html字符串和语法获取链接地址

private static String selectDocumentLink(String htmlText,String Query) {

Document doc=Jsoup.parse(htmlText);

String select=doc.select(Query).attr("href");

return select;

}

@Override

public Site getSite() {

return Site.me().setSleepTime(1000).setRetryTimes(10);

}

//编写抽取逻辑

@Override

public void process(Page page) {

page.addTargetRequests(urls());

//定义如何抽取页面信息

List<String> htmls=page.getHtml().xpath("//div[@class='post\_item']/html()").all();

List<JavaBokeModel> javaBokes=new ArrayList<JavaBokeModel>();

for(String html:htmls) {

JavaBokeModel javaBoke=new JavaBokeModel();

//标题和链接

String title=selectDocumentText(html,TITLEQUERY);

String linke=selectDocumentLink(html,TITLEQUERY);

//作者和作者主页

String author=selectDocumentText(html,AUTHORQUERY);

//简介

//System.out.println(title);

javaBoke.setTitle(title);

javaBoke.setAuthor(author);

javaBoke.setLinke(linke);

javaBokes.add(javaBoke);

}

File.WriteStringToFile2(javaBokes);

}

public static void main(String[] args) {

long startTime,endTime;

DBUtil.getConnection();

startTime=new Date().getTime();

Spider create=Spider.create(new BlogPageProcessor());

create.addUrl("http://www.cnblogs.com/").thread(5).run();

try {

ps.close();

conn.close();

}catch(Exception e) {

}

endTime=new Date().getTime();

System.out.println("用时为："+(endTime-startTime)/1000+"s");

}

}

package zuoye1;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import us.codecraft.webmagic.Page;

import us.codecraft.webmagic.Site;

import us.codecraft.webmagic.Spider;

import us.codecraft.webmagic.processor.PageProcessor;

public class CSDNPageProcessor implements PageProcessor {

private static Connection conn = null;

private static PreparedStatement ps = null;

// 标题和链接获取

private static String TITLEQUERY = "div.list\_con div.title h2 a";

// 作者

private static String AUTHORQUERY = "dl.list\_userbar dd.name a";

//初始化带爬取网页地址

private static List<String> urls() {

List<String> listUrl=new ArrayList<String>();

String[] type=new String[]{"nav/newarticles","nav/news","nav/ai","nav/cloud","nav/db","nav/career","nav/game","nav/engineering","nav/other"};

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

String url2="https://blog.csdn.net/"+type[i];

listUrl.add(url2);

}

return listUrl;

}

//jsoup根据html字符串和语法来获取内容

private static String selectDocumentText(String htmlText,String Query) {

Document doc=Jsoup.parse(htmlText);

String select=doc.select(Query).text();

return select;

}

//jsoup根据html字符串和语法获取链接地址

private static String selectDocumentLink(String htmlText,String Query) {

Document doc=Jsoup.parse(htmlText);

String select=doc.select(Query).attr("href");

return select;

}

@Override

public Site getSite() {

return Site.me().setSleepTime(1000).setRetryTimes(10);

}

//编写抽取逻辑

@Override

public void process(Page page) {

page.addTargetRequests(urls());

//定义如何抽取页面信息

List<String> htmls=page.getHtml().xpath("//li[@class='clearfix']/html()").all();

List<JavaBokeModel> javaBokes=new ArrayList<JavaBokeModel>();

for(String html:htmls) {

JavaBokeModel javaBoke=new JavaBokeModel();

//标题和链接

String title=selectDocumentText(html,TITLEQUERY);

String linke=selectDocumentLink(html,TITLEQUERY);

//作者和作者主页

String author=selectDocumentText(html,AUTHORQUERY);

//简介

System.out.println(title+" "+linke+" "+author);

javaBoke.setTitle(title);

javaBoke.setAuthor(author);

javaBoke.setLinke(linke);

javaBokes.add(javaBoke);

}

File.WriteStringToFile2(javaBokes);

}

public static void main(String[] args) {

long startTime,endTime;

DBUtil.getConnection();

startTime=new Date().getTime();

Spider create=Spider.create(new CSDNPageProcessor());

create.addUrl("https://blog.csdn.net/").thread(5).run();

try {

ps.close();

conn.close();

}catch(Exception e) {

}

endTime=new Date().getTime();

System.out.println("用时为："+(endTime-startTime)/1000+"s");

}

}

1. **爬取热词解释源代码**

package zuoye1\_2;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import us.codecraft.webmagic.Page;

import us.codecraft.webmagic.Site;

import us.codecraft.webmagic.Spider;

import us.codecraft.webmagic.processor.PageProcessor;

public class MingciPangProcessor implements PageProcessor{

public static List<String> titles;

public static List<mean> mean2=new ArrayList<mean>();

public static int i=0;

public static List<String> words=new ArrayList<String>();

@Override

public Site getSite() {

return Site.me().setSleepTime(1000).setRetryTimes(10);

}

//jsoup根据html字符串和语法来获取内容

private static String selectDocumentText(String htmlText) {

Document doc=Jsoup.parse(htmlText);

String select=doc.text();

return select;

}

//初始化带爬取网页地址

private static List<String> urls() {

List listUrl=new ArrayList<String>();

titles=readFile.toArrayByInputStreamReader1();

for(String title:titles) {

listUrl.add("https://baike.baidu.com/item/"+title);

words.add(title);

}

listUrl.toArray(new String[listUrl.size()]);

return listUrl;

}

@Override

public void process(Page page) {

page.addTargetRequest(urls().get(i));

//定义如何抽取页面信息

List<String> htmls=page.getHtml().xpath("//div[@class='lemma-summary']/html()").all();

for(String html:htmls) {

String mean=selectDocumentText(html);

String word=words.get(i);

System.out.println(word+":"+mean);

mean mean1=new mean();

mean1.setMeann(mean);

mean1.setWord(word);

mean2.add(mean1);

}

i=i+1;

}

public static void main(String[] args) {

long startTime,endTime;

startTime=new Date().getTime();

Spider create=Spider.create(new MingciPangProcessor());

words.add("信息");

String strr="https://baike.baidu.com/item/信息";

create.addUrl(strr).thread(5).run();

endTime=new Date().getTime();

File.WriteStringToFile2(mean2);

System.out.println("用时为："+(endTime-startTime)/1000+"s");

}

}

1. **文章标题切词源代码**

import jieba  
*#读取文件*f=open(r"F:\大数据\大作业\爬取到的数据\data1\_xinxi.txt",'r')  
s=f.read()  
*#print(s)  
#切割文件中的字符串*zifuchuan=s.split("\n");*#按行分割*i=0  
zifuchuan1=[]*#标题*zifuchuan2=[]*#文章链接*zifuchuan3=[]*#作者*for ss in zifuchuan:  
 if ss!='':*#去掉空行  
 #print(":"+ss)* zifu=ss.split("\t")  
 zifuchuan1.append(zifu[0])  
 zifuchuan2.append(zifu[1])  
 zifuchuan3.append(zifu[2])  
*#print(zifuchuan1)  
#分词*k1=0  
k2=0  
fencihou2=[]  
for ss in zifuchuan1:  
 *#print(ss)* fencihou=jieba.lcut(ss,cut\_all=True)  
 *#print(fencihou)* for f in fencihou:  
 if f.isspace()==False:  
 print("@:"+f)  
 fencihou3 = f + ' ' + zifuchuan2[k2] + ' ' + zifuchuan3[k2]  
 fencihou2.append(fencihou3)  
 k1 = k1 + 1  
  
 k2=k2+1  
*#print(fencihou2)  
#写入文件*f1=open("F:\大数据\大作业\分词后的文件\data2\_xinxi.txt",'a+')  
for fencihou4 in fencihou2:  
 f1.write(fencihou4)  
 f1.write("\n")  
f1.close()

1. **热词清洗源代码**

*#创建停用词表*def stopwordlist():  
 stopwords=[line.strip() for line in open ('F:\大数据\大作业\分词后的文件\stopWord.txt','r').readlines()]  
 return stopwords  
  
f=open(r"F:\大数据\大作业\分词后的文件\data2\_xinxi.txt",'r')  
s=f.read()  
*#切割文件中的字符串*zifuchuan=s.split("\n");*#按行分割*i=0  
stopwords=stopwordlist()  
zifuchuan2=[]*#过滤后的信息*zifuchuan3=[]  
for ss in zifuchuan:  
 zifu=ss.split(" ")  
 print(":"+zifu[0])  
 if zifu[0].isdigit()==False:  
 if zifu[0] not in stopwords:  
 zifuchuan2.append(ss)  
 zifuchuan3.append(zifu[0])  
  
*#写入文件*f1=open("F:\大数据\大作业\分词后的文件\data3\_xinxi.txt",'a+')  
for z in zifuchuan2:  
 f1.write(z)  
 f1.write("\n")  
f1.close()  
f2=open("F:\大数据\大作业\分词后的文件\data4\_xinxi.txt",'a+')  
for z1 in zifuchuan3:  
 f2.write(z1)  
 f2.write("\n")  
f2.close()

1. **统计热词源代码**

def main():  
 file=open("F:\大数据\大作业\分词后的文件\data4\_xinxi.txt",'r')  
 wordCounts={} *#先建立一个空的字典，用来存储单词 和相应出现的频次* count=50 *#显示前多少条（按照单词出现频次从高到低）* for line in file:  
  
 lineprocess(line.lower(),wordCounts) *#对于每一行都进行处理，调用lineprocess()函数，参数就是从file文件读取的一行* items0=list(wordCounts.items()) *#把字典中的键值对存成列表，形如：["word":"data"]* items=[[x,y] for (y,x) in items0] *#将列表中的键值对换一下顺序，方便进行单词频次的排序 就变成了["data":"word"]* items.sort() *#sort()函数对每个单词出现的频次按从小到大进行排序* gailv=[]*#存储像文件中输入的词语以及概率* for i in range(len(items)-1,1,-1): *#上一步进行排序之后 对items中的元素从后面开始遍历 也就是先访问频次多的单词* if items[i][0]<10:  
 break;  
 zz=items[i][1]+"\t"+str(items[i][0])  
 gailv.append(zz)  
 f2 = open("F:\大数据\大作业\分词后的文件\data5\_xinxi.txt", 'a+',encoding='utf-8')  
 for z1 in gailv:  
 f2.write(z1)  
 f2.write("\n")  
 f2.close()  
  
  
def lineprocess(line,wordCounts):  
 for ch in line: *#对于每一行中的每一个字符 对于其中的特殊字符需要进行替换操作* if ch in "~@#$%^&\*()\_-+=<>?/,.:;{}[]|\'""":  
 line=line.replace(ch,"")  
 words=line.split() *#替换掉特殊字符以后 对每一行去掉空行操作,也就是每一行实际的单词数量* for word in words:  
 if word in wordCounts:  
 wordCounts[word]+=1  
 else:  
 wordCounts[word]=1  
  
 *#这个函数执行完成之后整篇文章里每个单词出现的频次都已经统计好了*main()