|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **北京邮电大学软件学院** | **文档编号** | TDIPS-4 | **版本** | 4.0 | **密级** | 商密A |
| **项目名称** | **文本数据智能处理系统** | | | | |
| **项目来源** |  | | | | |

**文本数据智能处理系统**

**概要设计说明书**

(内部资料请勿外传)

|  |  |  |  |
| --- | --- | --- | --- |
| **编写：** | 刘凡凡 林哲  吴晓吟 任建新 | **日期：** | 2011-1-10 |
| **检查：** | 任建新 | **日期：** | 2011-1-11 |
| **审核：** | 吴国仕 | **日期：** | 2011-1-11 |
| **批准：** |  | **日期：** |  |

**北京邮电大学软件学院**

**版权所有不得复制**

**文档变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 | 初次编写 | 任建新 | 1.0 | 2011-1-12 |  |
| 2 | 增加AJAX爬虫模块概要设计 | 刘凡凡 | 2.0 | 2011-1-14 |  |
| 3 | 增加情感化过滤模块概要设计 | 吴晓吟 | 3.0 | 2011-1-14 |  |
| 4 | 增加网页数据精确抽取模块概要设计 | 林哲 | 4.0 | 2011-1-14 |  |
|  |  |  |  |  |  |

目录

[1. 引言 6](#_Toc316718935)

[1.1. 编写目的 6](#_Toc316718936)

[1.2. 项目背景 6](#_Toc316718937)

[1.3. 定义及缩略语 7](#_Toc316718938)

[1.4. 参考资料 8](#_Toc316718939)

[1.5. 使用的绘图工具 8](#_Toc316718940)

[2. 任务概述 9](#_Toc316718941)

[2.1. 目标 9](#_Toc316718942)

[2.2. 运行环境 9](#_Toc316718943)

[2.3. 需求概述 9](#_Toc316718944)

[2.4. 条件与限制 10](#_Toc316718945)

[3. 支持AJAX的定址网络爬虫模块 11](#_Toc316718946)

[3.1. 系统概要设计 11](#_Toc316718947)

[3.1.1. 系统体系结构 11](#_Toc316718948)

[3.1.2. 系统功能模块 11](#_Toc316718949)

[3.1.2.1. 浏览器 11](#_Toc316718950)

[3.1.2.2. Cookie管理 12](#_Toc316718951)

[3.1.2.3. 通信模块 12](#_Toc316718952)

[3.1.3. 基本设计概念和处理流程 12](#_Toc316718953)

[3.1.3.1. 爬虫工作流程 12](#_Toc316718954)

[3.1.3.2. 浏览器工作流程 13](#_Toc316718955)

[3.2. 接口设计 14](#_Toc316718956)

[3.2.1. 用户接口 14](#_Toc316718957)

[3.2.2. 外部接口 14](#_Toc316718958)

[3.2.3. 内部接口 14](#_Toc316718959)

[3.3. 出错处理设计 14](#_Toc316718960)

[3.3.1. 出错信息与处理对策 14](#_Toc316718961)

[3.3.2. 模块维护设计 14](#_Toc316718962)

[4. 网页数据精确抽取模块 15](#_Toc316718963)

[4.1. 总体设计 15](#_Toc316718964)

[4.1.1. 处理流程 15](#_Toc316718965)

[4.1.2. 总体结构和模块设计 15](#_Toc316718966)

[4.2. 接口设计 16](#_Toc316718967)

[4.2.1. 外部接口 16](#_Toc316718968)

[4.2.2. 内部接口 17](#_Toc316718969)

[4.3. 数据结构设计 17](#_Toc316718970)

[4.3.1. 模块数据结构 17](#_Toc316718971)

[4.3.2. 数据结构与程序的关系 18](#_Toc316718972)

[4.4. 运行模块的组合 19](#_Toc316718973)

[4.5. 出错处理设计 19](#_Toc316718974)

[4.5.1. 出错输出信息 19](#_Toc316718975)

[4.5.2. 出错处理对策 19](#_Toc316718976)

[5. 情感化过滤模块 20](#_Toc316718977)

[5.1. 总体设计 20](#_Toc316718978)

[5.1.1. 处理流程 20](#_Toc316718979)

[5.1.2. 总体结构 22](#_Toc316718980)

[5.2. 接口设计 22](#_Toc316718981)

[5.2.1. 外部接口 22](#_Toc316718982)

[5.2.2. 内部接口 23](#_Toc316718983)

[5.3. 结构设计 24](#_Toc316718984)

[5.3.1. 逻辑结构设计 24](#_Toc316718985)

[5.3.1.1. 数据字典 24](#_Toc316718986)

[5.3.1.2. 数据结构 28](#_Toc316718987)

[5.3.2. 物理结构设计 28](#_Toc316718988)

[5.3.2.1. 数据字典 28](#_Toc316718989)

[5.3.2.2. 数据结构 31](#_Toc316718990)

[5.3.3. 数据结构与程序的关系 32](#_Toc316718991)

[5.4. 运行模块的组合 33](#_Toc316718992)

[5.5. 出错处理设计 33](#_Toc316718993)

[5.5.1. 出错输出信息 33](#_Toc316718994)

[5.5.2. 出错处理对策 33](#_Toc316718995)

[6. 维护设计 33](#_Toc316718996)

# 引言

## 编写目的

由前面的需求分析，得出了系统的基本需求，和基本的数据流图，要实现整个系统，需要对用户的需求进行设计，概要设计主要是利用比较抽象的语言对整个需求进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。

本文档给出了文本数据智能处理系统的概要设计说明，包括系统的结构、基本处理流程，最终实现的软件必须满足的功能、性能和接口等，为系统的详细设计提供基础。本说明书的编写目的在于：为编码人员提供依据：为修改、维护提供条件。本说明书的预期读者包括项目开发人员、软件维护人员、合作各方有关部门的负责人、项目组负责人和全体参加人员。

## 项目背景

文本数据智能处理系统是一个比较大的系统，根据用户需求，我们将系统划分为主要的3大模块，分别是支持AJAX的定址网络爬虫模块，网页数据精确抽取模块和情感化过滤模块。

随着互联网内容爆炸性增长，利用网页信息提取技术和数据挖掘技术采集情报变得越来越重要，AJAX技术由于符合Web2.0时代的需要，一经提出就引起了互联网领域的广泛关注，它的问世将传统的同步式页面转变为异步式页面，给用户带来了桌面式的网页体验。但同时这一技术给搜索引擎带来了巨大的挑战。异步页面，事件触发的模式使得传统的搜索引擎前端（即网络爬虫）无法获取到全部的网页信息。通过分析和研究现有的网络爬虫和网站信息抓取软件工具，分析比较其中的优缺点和功能限制，在此基础上，形成了支持AJAX的定址网络爬虫。

信息精确抓取的目标是帮助用户从网上获取对他们真正有价值的信息。当前的web信息抽取系统，诸如baidu，google等搜索引擎或firtex等垂直搜索系统只能为信息需求者提供一个庞大的结果集，具体的信息还是要用户到具体的网页中查找。这种方式会耗费用户大量的时间，并且提供的结果也未必是用户真正感兴趣的内容。因此，通过对WEB信息的获取，抽取对用户真正有价值的信息，是提高企业竞争力，帮助企业员工或政府职能部门公务人员获取信息，掌握社会发展动态，了解世界状况的有效方法。

互联网用户乐于在网络上分享自己的心情、观点与体验，因此，互联网（如Weblog和BBS论坛）上产生了大量的反映用户个体感情色彩的信息。但是，这些互联网用户产生的海量信息中夹杂着很多不健康的内容，这些不健康的内容很有可能误导公众的价值观，扰乱社会的和谐与稳定。因此，加强舆论的监管对于当今社会信息安全和维稳有着不可忽视的作用。

* 本系统的名称：

文本数据智能处理系统

* 任务提出者：

卓望公司

* 开发商：

北京邮电大学软件学院

* 硬件环境：

PC机一台

## 定义及缩略语

* Web2.0

Web2.0[佚名，2005]是相对Web1.0（2003年以前的互联网模式）的新的一类互联网应用的统称，是以Flickr、Craigslist、Linkedin、Tribes、Ryze、Friendster、Del.icio.us、43Things.com等网站为代表，以Blog、TAG、SNS、RSS、wiki等应用为核心，依据六度分隔、XML、AJAX等理论和技术实现的互联网新一代模式。

* AJAX

AJAX（Asynchronous JavaScript and XML）是一种用于创建快速动态网页的技术，通过在后台与服务器进行少量数据交换，AJAX可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

* JavaScript/Js

JavaScript是因特网上最流行的脚本语言，并且可在所有主要的浏览器中运行，比方说 Internet Explorer、Mozilla、Firefox、Netscape和Opera，它被用来改进设计、验证表单、检测浏览器、创建cookies等。

表1-1术语或缩略词列表

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **术语或缩略词** | **说明性定义** | **对应功能** |
| 1 | AJAX爬虫 | 为支持AJAX的定址网络爬虫的缩写 | 支持AJAX的定址网络爬虫 |
| 2 | 红黑词典 | 用于情感分析的词典 | 情感化过滤 |

## 参考资料

[1] 支持AJAX的网络爬虫系统设计与实现曾伟辉中国科学技术大学,2009

[2] 《软件工程导论——第5版》张海藩编著清华大学出版社

[3] 《软件工程实践导论：有关方法、设计、实现、管理之三十六计》金尊和编著清华大学出版社

[4] 《实用软件工程》Leszek，A.Maciaszek，Bruc Lee Liong著机械工业出版社

[5] L. Arllota, V. Crescenzi, G. Mecca, and P. Merialdo.Automatic annotation of data extraction from large Web sites. In Proceedings of the International Workshop on the Web and Databases, pages 7–12, San Diego, USA, 2003.[6] S. Zheng, D. Wu, and R. Song.Joint Optimization of Wrapper Generation and Template Detection. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007

[7] Nitin Jindal and Bing Liu. Identifying Comparative Sentences in Text Documents[J]. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR-06). 2006.

[8]王春辉. “假设性等级”与汉语条件句[J].《汉语学报》2010年第04期.

[9] Nitin Jindal and Bing Liu. Identifying Comparative Sentences in Text Documents[J]. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR-06). 2006.

[10]Fei Xia. The Part-Of-Speech Tagging Guidelines for the Penn Chinese TreeBank(3.0). 2000.

## 使用的绘图工具

Microsoft Office Visio 2007

# 任务概述

## 目标

文本数据智能处理系统主要分为3个模块，他们是支持AJAX的定址网络爬虫模块、网页数据精确抽取模块以及情感化过滤模块。他的主要功能是通过AJAX动态页面爬取技术扩大爬取范围，并对抓取后的网页进行模板训练和信息抽取，最后对抽取后的信息进行句子以及文档级别的情感分析。

通过该系统的实施，AJAX爬虫将包括动态页面爬取后交给网页数据精确抽取模块，抽取模块将相同结构的网页通过生成的抽取模板、信息标注操作后，精确的抽取出用户感兴趣的网页数据。为后续的情感分析、信息搜集等模块提供数据。情感分析模块则利用计算机对带有文本作者主观情感或意见的文本进行自动的分析，并推理出其情感倾向性。输入的文本可以是句子或者文档，输出为文本的情感得分，正面文档得分为正，负面得分为负。

## 运行环境

运行平台：Windows XP/Windows 2000/Windows 2003/Win7

JDK1.6

## 需求概述

该文本数据智能处理系统主要包括3个功能，他们分别为支持AJAX的定址网络爬虫，网页数据精确抽取和情感化过滤。

支持AJAX的定址网络爬虫是解决从层出不穷的异步交互式网络的AJAX框架网站上抓取页面和翻页抓取问题的方案，主要解决基于Web2.0模式的采用AJAX技术的互联网应用的页面信息爬取。

网页数据精确抽取则是对爬取网页的进一步处理，主要包括生成模板，模板标注以及数据抽取3个功能。

情感化过滤则利用计算机对带有文本作者主观情感或意见的文本进行自动的分析，并推理出其情感倾向性。输入的文本可以是句子或者文档，输出为文本的情感得分，正面文档得分为正，负面得分为负。

## 条件与限制

* **支持AJAX的定址网络爬虫：**

1. 限制AJAX爬虫正常工作的主要因素是基础数据（如网页集文件）的完整性和翻页脚本的准确性。另外，由于各个网站网页的结构不同，在维护的同时，要根据特殊网页编写相应的处理方法。
2. 限制AJAX爬虫的另外一个因素是系统的维护性和完善性，因为爬虫系统的建立不可能一次全部解决问题，开发后期和使用初期的维护工程也是一个必须考虑的因素。这里的维护工作包括数据的维护，也有系统功能的维护，包括旧功能的删除和新功能的添加等。

* **情感化过滤：**

情感化过滤在正确安装JDK的机器上即可成功运行，但情感的褒贬义分析中影响其分析正确性的限制因素主要有以下几点：

1. 涵盖尽可能广的范围的尽可能准确的词典对情感分析判断的结果有着不可忽视的重要作用。本系统中使用的红黑词典是通过人工收集的，收集的主题词和情感词虽然已包含较多词汇量，但因为种种原因，如新词汇的出现或者中文词语含义的多义性，导致词典仍然不够完善，部分带有情感倾向的情感词并没有能够识别。所以，红黑词典的完善程度对最终准确度的高低有着较大的限制。
2. 限制本系统运行精度的另一个原因是篇幅大小问题，如一篇文章若篇幅过小，其中主观句的数量就相对较少，很有可能一两句话就是判断篇章情感倾向的依据，而句子中包含情感倾向的主题词和情感词过少，导致了判断的误差。
3. 还有一个不可忽视的限制因素是系统的维护性与完善性。因为中文句式的复杂与特殊，系统的在开发过程中很难一步到位地解决所有句式的分析问题。系统开发后期和使用初期的维护过程也是必须考虑的因素。这里的维护包括数据的维护和系统功能的维护等。
4. 由于情感分析主要用于分析句子的褒贬性质，对新闻类及相关网站内容的支持相对较好。如果对其他性质的网站内容进行分析处理，需要对其建立相应的词典来完善。

# 支持AJAX的定址网络爬虫模块

## 系统概要设计

### 系统体系结构



图3-1 支持AJAX的网络爬虫系统架构图

AJAX框架网络爬虫系统的核心是浏览器引擎，采用Qt Webkit技术实现，与Cookie管理模块交互形成一个简易的浏览器，执行网页的分析、渲染与呈现。同时浏览器使用Url资源库、翻页脚本和配置文件等作为输入，在爬行的过程中输出工作日志。Cookie管理模块利用Cookie文件实现Cookie的保存和读取，并在浏览器工作过程中管理Cookie信息。用户界面分为程序主界面和浏览器界面，方便用户管理网页抓取过程、监控网页抓取状态、观察网络爬虫所到网页等等。

### 系统功能模块

爬虫系统将浏览器引擎集成进爬虫内部，构建了一个浏览器，作为爬虫的核心模块，除此之外，还包括链接抽取模块、脚本生成模块、批量抓取模块及为浏览器提供Cookie支持的Cookie模块。

#### 链接抽取模块

链接抽取模块用来抽取一定数量的Url形成一个Url集的txt文件。

#### 脚本生成模块

生成脚本模块主要用来针对同一类结构的网页生成翻页脚本文件。

#### 批量抓取模块

批量抓取模块主要使用Url资源库、翻页脚本和配置文件等作为输入，利用浏览器批量爬行链接，获取页面内容，并写入本地静态html文件。形成AJAX站点的本地镜像。

#### 浏览器

浏览器的主要功能是加载网页，分析页面，解析JavaScript代码和AJAX代码，当网页加载完成后，保存页面信息，并执行对应网站的翻页规则脚本，获取分页内容。

#### Cookie管理

对Cookie进行管理，主要包括读取、保存和清除操作。

### 基本设计概念和处理流程

本系统是一个定址网络爬虫，爬行范围完全基于Url文件驱动，哪些超链接符合主题由操作员告知网络爬虫，指导其抓取网站信息，简而言之，就是爬行的范围是受控的。它采用基于DOM+JavaScript的独特的网站抓取策略，使用直观的图形化界面，内核是Webkit网页解析引擎，并能够利用翻页脚本文件，连续地或者周期性地抓取网站信息，并将信息以HTML网页格式存储在电脑磁盘上。

#### 爬虫工作流程

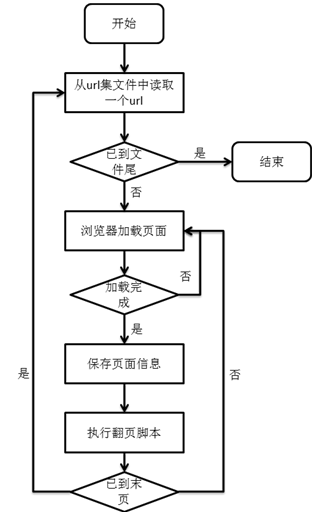


图3-2 爬虫工作流程

爬虫整体工作流程描述如下：

1. 爬虫程序运行后，初始化浏览器，读取网页集中的第一个网址，并加载网页。
2. 网页加载完成后，爬虫向浏览器发出加载完成的信号，浏览器得知网页加载完成，执行输入项中对应的翻页脚本。
3. 翻页脚本执行翻页操作，直至到达尾页，则翻页结束。在每翻一页之前，都向浏览器发出保存网页信号，浏览器接收到该信号，保存当前页面信息。
4. 翻页结束后，浏览器读取网页集中当前网址的后一个，并加载网页，程序转向第2步，直到将网页集中的所有网址都遍历完毕，爬行过程结束。
5. 浏览器保存Cookie，执行相关清理工作，关闭窗口。

#### 浏览器工作流程

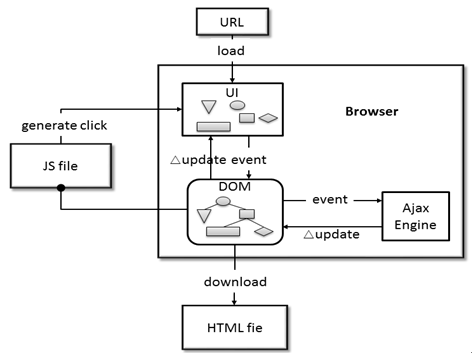


图3-3 浏览器工作流程

浏览器详细工作流程描述如下：

浏览器首先根据指定的URL获取网页，网页加载过程中，通过Ajax引擎与远程服务器进行异步数据交换，动态更新页面的DOM树结构，DOM树结构的变化引起网页内容与结构的变化，当网页加载完成后，浏览器将当前网页内容保存在电脑磁盘上，并在浏览器上下文环境中执行翻页脚本，翻页脚本向指向翻页的DOM节点发送用户点击消息，触发JavaScript/Js代码，在代码中使用XMLHttpRequest等通信函数下载HTML分页页面。接着浏览器按照加载、保存、翻页的顺序循环执行，直至翻页结束。

## 接口设计

### 用户接口

提供直观的图形化用户界面，方便用户管理网页抓取过程、监控网页抓取状态、观察网络爬虫所到网页等等。

### 外部接口

外部程序可通过为浏览器提供网页集文件、翻页脚本文件和抓取下来的网页的存储目录直接调用浏览器，执行网页抓取。

### 内部接口

在浏览器模块中可以使用通信模块，在用户界面中输出爬行信息和出错信息。

在浏览器模块中使用Cookie管理模块实现对Cookie的管理。

## 出错处理设计

### 出错信息与处理对策

* 打开crawler.jar文件时报错，提示“Could not find the main class”的警告。

出现这种错误可能是由与Jar文件关联的程序不正确导致的，解决办法请参考：http://blog.csdn.net/wuxianglong/article/details/5894236

* 登录网易微博没反应。

请安装对应操作系统版本的OpenSSL，完整版或简易版皆可，并将安装路径加入系统变量PATH中，Windows系统的下载地址请访问http://www.slproweb.com/products/Win32OpenSSL.html。

### 模块维护设计

维护方面主要是对翻页规则脚本文件进行维护。当网页结构发生变化后，需要人工修改翻页规则文件以适应新的网站结构。

# 网页数据精确抽取模块

## 总体设计

### 处理流程

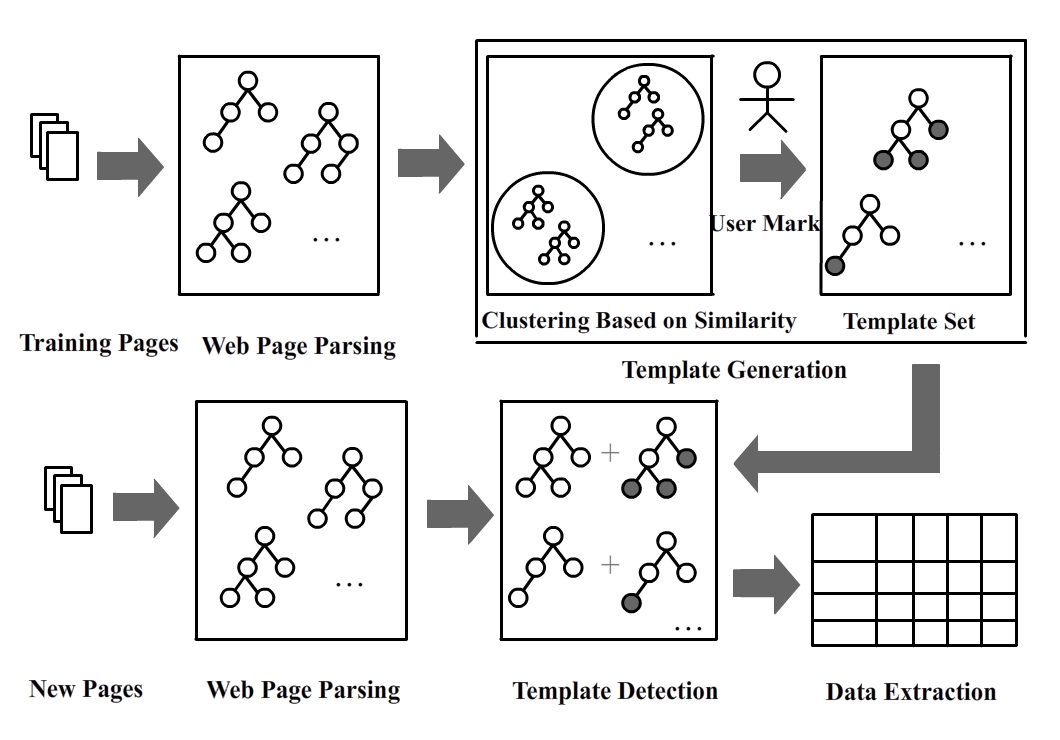


图4-1 数据抽取模块总体处理过程

如图4-1所示，通过计算机对网页集进行训练，生成网页模板，用户通过模版选取感兴趣的信息，对其进行标记（语义信息）。利用用户标记过的模板，对指定的网页抽取出用户感兴趣的数据。对于待抽取的网页，根据该网页和模板库中的模板的相似度，选择对应（相似度最高）的模板，抽取出用户感兴趣的数据。抽取的信息可以按用户需要放到指定的文件（XML，HTML）或者用户数据库中。抽取的数据可以作为数据挖掘、数据分析等其他系统的数据基础，因此需要为用户提供较为丰富的数据保存形式

### 总体结构和模块设计

针对网页数据精确抽取系统，可以分为三个模块：分别为生成模板、模板标注、数据抽取；如图4-2所示：



图4-2 数据抽取模块系统结构

* 生成模板模块

用户将具有相同网页结构的网页集作为输入，选择相似度后生成模板。

* 模板标注模块：

该模块主要是处理由生成模板模块生成的网页模板，并标注用户感兴趣的网页信息，为数据抽取做准备。在操作人员选择所要进行标注的模板文件，双击选择所要标注的信息，在菜单上选择相应的操作，标注操作结束后，将标注文件保存后退出。

* 数据抽取模块：

该模块主要是抽取新输入的网页集，在操作人员选择标注好的模板文件和所要进行抽取的网页集后，点击相应的操作按钮后对网页进行抽取。

## 接口设计

### 外部接口

* 用户接口：

采用窗口化，菜单式进行设计。

* 软件接口

根据用户需要，可以将抽取数据通过JDBC对MYSQL数据库进行存储，或者将生成的数据以xml的格式存储。

外部程序可以通过类Extract中的extract方法调用实现生成模板的功能。

调用方式

|  |
| --- |
| Extract extractData= **new** Extract();  extractData.setTempalteFile(模板文件名);  extractData.setDestDirectory(结果存放目录);  extractData.setExtractFiles(FileHelp.*getFiles*(网页集目录);  extractData.setStatusBar(statusBar);  extractData.setExtractType(Extract.*NORMAL*);  extractData.extract(); |

### 内部接口

通过面向对象语言设计类，在public类中实现调用；类间实现严格封装；

## 数据结构设计

### 模块数据结构

模块的数据结构见表4-1至4.3。

表4-1 标注数据类（MarkData）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 字段名 | 类型 | 说明 |
| 1 | semantic | 标注 | String | 标注名称 |
| 2 | block | 块选择名称 | String | 块选择名称 |
| 3 | windowStatus | 标签编号 | String | 对应的标签编号 |

表4-2 抽取数据内部类 (WieData)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 字段名 | 类型 | 说明 |
| 1 | semantic | 图书编号 | Map<String, String> | 标注的名称 |
| 2 | block | 条形码 | Map<String,List<String>> | 标注的标签对应的块选择的值列表 |

表4-3 抽取数据类 (ExtractData)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 字段 | 字段名 | 类型 | 说明 |
| 1 | titles | 抽取数据的名称 | List<String> | 抽取数据的名称 |
| 2 | datas | 抽取数据的元数据 | Map<String,List<Element>> | 抽取数据的元数据 |
| 3 | WieData | 内部类 | WieData |  |

### 数据结构与程序的关系

数据结构与程序是软件的重要组成部分，程序的正确执行依赖于合理的数据结构。在4.1中定义的类中都位于包com.mywie.model中，在com.mywie.core中的Extract类中被调用。

运行设计。

## 运行模块的组合

本程序主要是以一个窗口为模块，一般一个窗口完成一个特定的功能，主窗口通过打开另一个子窗口来实现个模块之间不同功能的连接和组合。各模块之间相对独立，程序的可移植性好。各模块之间主要以传递数据项的引用来实现模块之间的合作和数据共享。

## 出错处理设计

### 出错输出信息

模块多处采用异常处理的机制，当遇到异常时不但能及时的处理，保证程序的安全性和稳定性，而且各种出错信息能通过弹出对话框的形式，及时的告诉用户出错的原因及解决的办法，使用户以后能够减少错误的发生。

### 出错处理对策

对于模块的几种可能出现的错误，分别给出相应的处理。

主要的错误可能有：

* 输入错误：

这主要是用户输入不规范造成的，我们在尽量减少用户出错的条件的情况下，主要也是通过对话框，提醒用户，然后再次操作。

* 其他操作错误：

对于用户的不正当操作，有可能使程序发生错误。我们主要是中止操作，并提醒用户中止的原因和操作的规范。

* 其他不可预知的错误：

程序也会有一些我们无法预知或没考虑完全的错误，我们对此不可能作出万全的异常处理，这时我们主要要保证数据的安全，所以要经常的进行数据库备份，并能及时的和我们联系，以逐步的完善我们的程序。

# 情感化过滤模块

## 总体设计

### 处理流程

* 顶层数据流程如图5-1所示：



图5-1 顶层数据流程图

* 一层数据流程如图5-2所示：



图5-2 一层数据流程图

* 句子级情感分析二级数据流程如图5-3所示：



图5-3 句子级情感分析二级数据流程图

* 文档级情感分析二级数据流程如图5-4所示：



图5-4 文档级情感分析二级数据流程图

### 总体结构



图5-5 情感化过滤模块结构图

从下到上依次介绍：

最下面一层是语料数据层，保存词典与各模式库内容。

中间一层是处理层，这是本系统最为核心的一块内容，主要包括预处理功能、复合句处理功能、比较句处理功能、简单句处理功能和情感得分计算功能。

上面一层是服务层，主要是为用户提供情感分析的服务，包括句子级的情感分析和文档级情感分析。

## 接口设计

### 外部接口

* 用户接口：

为了方便用户测试及使用，模块提供了窗口化，菜单式的设计，在操作时响应热键。

* 模块外部接口：

分词工具：ICTCLAS2010类（ICTCLAS\_Paragraphprocess方法）

句法分析工具接口：LexicalizedParser类

Livsvm工具接口：svm\_predict类（public static boolean runSVM（String keyword）方法）

### 内部接口

通过面向对象语言设计类，在public类中实现调用；类间实现严格封装

句子级别的情感分类模型有统一的处理过程，封装的API参数接口也是一致的，分别如下：

句法分析模型：

public LinkedList<SentiResult> syntaxAnalyze(String review);

基于向量空间分析模型:

public LinkedList<SentiResult> windowAnalyze(String review);

强力模型：

public LinkedList<SentiResult> simpleAnalyze(String review);

篇章级的接口实现用表给出：

系统接口实现

|  |
| --- |
| //分词接口  String newstr1=WordSplitter（ReiewAll[nLength]);  //复合句型处理接口  LinkedList<XMLResult> res = new LinkedList<XMLResult>();  //复合句规则匹配接口，path即为条件句模式库的文件路径  Search search1 = new Search();  res = search1.identityComplex(newstr1 ,path);  //复合句型情感分析接口  ReviewAll[nLength] =ProcessComplexSentence.processComplexSentence(ReviewAll[nLength], res.get(i).type);  //第二步，以逗号等分句块为单位切分并预处理，接下去进行比较句处理  //是否含有比较关键字  boolean haveKeyword = search.haveComparativeKeyword (newstr);  if(haveKeyword)  {  //判定是否为比较句  LinkedList<SVMResult> iscomparative  = search.indentifyComparativeBySVM (newstr);  //是比较句，进行情感分析  if(iscomparative.size()>0){  result2 =ProcessComparativeSentence.processcomparativesentence(iscomparative.get(i).keyword, newstr1);  //结果输出  }  //不是比较句的情况  Else{  //情感分析,基于语义指向的情感分析模型  result1 = ps.syntaxAnalyzeForSemantic(Review[counter]);  //结果输出  }  } |

## 结构设计

### 逻辑结构设计

#### 数据字典

表5-1 主题词表（topics）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 主题词序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 主题词 | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | 主题词情感极性 | Int | 10 |  |  |  | √ |  |  |

表5-2 情感词表（sentiment\_words）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 情感词序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 情感词 | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | 情感词情感极性 | Int | 10 |  |  |  | √ |  |  |

表5-3 否定词表（negations）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 否定词序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 否定词 | nvarchar | 50 |  |  |  | √ |  |  |

表5-4 二义词表（action\_words）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 二义词序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 二义词 | nvarchar | 50 |  |  |  | √ |  |  |

表5-5 主观句识别模式表(patterns)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 主观句模板序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 主观句模板 | nvarchar | 50 |  |  |  | √ |  |  |

表5-6 比较词表（compare\_words）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 比较词序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 比较词 | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | 比较词极性 | Int | 10 |  |  |  | √ |  |  |

表5-7 比较关键词表(compare\_keywords)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 比较关键词序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 比较关键词 | nvarchar | 50 |  |  |  | √ |  |  |

表5-8 比较句规则表(compare\_rules)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 比较规则序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 比较句规则 | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | 比较句规则类别 | Int | 10 |  |  |  | √ |  |  |

表5-9 比较对象表（compare\_objects）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 比较对象序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 比较对象 | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | 比较对象情感极性 | Int | 10 |  |  |  | √ |  |  |

表5-10 复合句规则表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | 复合句规则序号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | 复合句规则 | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | 复合句规则类别 | Int | 10 |  |  |  | √ |  |  |

#### 数据结构

1. **句法路径的步Step**

包含表示节点类型、节点的词性值和依赖关系树节点编号

1. **情感词基本数据结构类SentiNeg**

包含情感词和表示情感度正面、负面的权重度,只有+1、-1，还有表示情感词所在分句块中的位置，以及表示情感词前后否定词的次数权重，一次以1递增，默认为1

1. **用作的标注的词的结构类Word**

包含词和该词的词性标注

1. **用作情感分析的情感度数据结构SentiResult**

包含主题词，情感词和表示情感度正面、负面的权重度

1. **用作复杂句分析的数据结构类XMLResult**

包含规则和所属的类型

1. **用作比较句识别SVM模型的数据结构类SVMResult**

包含关键词和是否比较句，取值为true或者false

1. **用作比较句识别简化模型的数据结构类XMLResult**

包含规则和所属的类型

### 物理结构设计

#### 数据字典

表5-11 主题词表（topics）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | Polarity | Int | 10 |  |  |  | √ |  |  |

表5-12 情感词表（sentiment\_words）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | Polarity | Int | 10 |  |  |  | √ |  |  |

表5-13 否定词表（negations）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |

表5-14 二义词表（action\_words）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |

表5-15 主观句识别模式表(patterns)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Pattern | nvarchar | 50 |  |  |  | √ |  |  |

表5-16 比较词表（compare\_words）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id号 | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | Polarity | Int | 10 |  |  |  | √ |  |  |

表5-17 比较关键词表(compare\_keywords)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |

表5-18 比较句规则表(compare\_rules)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Rule | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | CategoryID | Int | 10 |  |  |  | √ |  |  |

表5-19 比较对象表（compare\_objects）

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Word | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | Polarity | Int | 10 |  |  |  | √ |  |  |

表5-20 复合句规则表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 长度 | 精度 | 小数位数 | 默认值 | 允许空 | 主键 | 说明 |
| 1 | Id | Int | 10 |  |  | 1 |  | √ | 自动编号 |
| 2 | Rule | nvarchar | 50 |  |  |  | √ |  |  |
| 3 | CategoryID | Int | 10 |  |  |  | √ |  |  |

#### 数据结构

1. **句法路径的步**

Step {

String dir; //表示节点类型、向上向下

String value; //节点的词性值

int id; //依赖关系树节点编号

}；

1. **情感词基本数据结构类**

SentiNeg {

String sentiment\_word; //情感词

int polarity; //表示情感度正面、负面的权重度,只有+1、-1

int pos; //表示情感词所在分句块中的位置

int power; //表示情感词前后否定词的次数权重，一次以1递增，默认为1

}；

1. **用作的标注的词的结构类**

Word {

String srcWord; //词

String tag; //标注

}；

1. **用作情感分析的情感度数据结构**

SentiResult {

String topic; //主题词

String sentiment\_word; //情感词

public int polarity; //表示情感度正面、负面的权重度

}；

1. **用作复杂句分析的数据结构类**

public class XMLResult {

public String rule; //规则

public int type; //类型

}

1. **用作比较句识别SVM模型的数据结构类**

public class SVMResult {

public String keyword; //关键词

public boolean iscomp; //是否比较句

}

1. **用作比较句识别简化模型的数据结构类**

public class XMLResult {

public String rule; //规则

public int type; //类型

}

### 数据结构与程序的关系

数据结构与程序是软件的重要组成部分，程序的正确执行依赖于合理的数据结构

## 运行模块的组合

本程序模块之间相对独立，程序的可移植性好。各模块之间主要以传递数据项的引用来实现模块之间的合作和数据共享。

## 出错处理设计

### 出错输出信息

本程序多处采用了异常处理的机制，当遇到异常时不但能及时的处理，保证程序的安全性和稳定性，而且各种出错信息能通过控制台打印，及时的告诉用户出错的原因及解决的办法，使用户以后能够减少错误的发生。程序的大部分地方还采取了出错保护，如输入内容的长度和类型等减少了用户出错的可能。

### 出错处理对策

我们对于本程序的几种可能的错误进行了分析，分别进行了不同的处理。

主要的错误可能有：

* 配置文件或相关文件如红黑词典等的导入错误：

这类错误主要是目标文件路径错误引起的，需要在指定的存储路径中存放相关文件。

* 输入错误：

这主要是用户输入不规范造成的，使用说明文档中应该详述输入格式。

* 其他操作错误：

对于用户的不正当操作，有可能使程序发生错误。使用说明文档中应该充分考虑操作规范。

* 其他不可预知的错误：

程序也会有一些我们无法预知或没考虑完全的错误，我们对此不可能作出万全的异常处理，这时我们主要要保证数据的安全，所以要经常的进行数据库备份，并能及时的和我们联系，以逐步的完善我们的程序。

# 维护设计

软件的维护主要包括，数据库的维护和软件功能的维护。

对于数据库的维护，主要是数据的更新与管理，本系统中数据库中的数据并不面向用户开放，故需要开发人员从后台进行数据维护。

对于软件功能方面的维护，由于我们采用的是模块化的设计方法，每个模块（窗口）之间相互独立性较高，这样对软件的维护带来了很大的方便。