**软件设计说明书**

**华南农业大学**

**数学与信息学院**

[1引言 2](#_Toc390206140)

[1.1编写目的 2](#_Toc390206141)

[1.2背景 2](#_Toc390206142)

[1.3定义 2](#_Toc390206143)

[1.4参考资料 3](#_Toc390206144)

[2总体设计 3](#_Toc390206145)

[2.1需求规定 3](#_Toc390206146)

[2.2运行环境 4](#_Toc390206147)

[2.3基本设计概念和处理流程 5](#_Toc390206148)

[2.4尚未解决的问题 9](#_Toc390206149)

[3接口设计 9](#_Toc390206150)

[3.1用户接口 9](#_Toc390206151)

[3.2外部接口 9](#_Toc390206152)

[3.3内部接口 10](#_Toc390206153)

[4运行设计 10](#_Toc390206154)

[4.1运行模块组合 10](#_Toc390206155)

[4.2运行控制 10](#_Toc390206156)

[4.3运行时间 10](#_Toc390206157)

[5系统数据结构设计 11](#_Toc390206158)

[5.1结构设计要点 11](#_Toc390206159)

[5.2数据类图 11](#_Toc390206160)

6 正文提取算法实现............................................................12

**设计说明书**

# 1引言

## 1.1编写目的

编写本说明书的目的是说明对程序系统的设计考虑，包括程序系统的处理流程、程序系统的组织结构、功能分配、模块划分、接口设计、运行设计、数据结构设计等。

## 1.2背景

说明：

1. 待开发的软件系统的名称：分布式爬虫系统
2. 本项目的任务提出者：“中国软件杯”大学生软件设计大赛组委会；
3. 本项目的任务开发者：华南农业大学 数学与信息学院。
4. 用户：“中国软件杯”大学生软件设计大赛评审委员会；

## 1.3定义

爬虫：是一种按照一定的规则，自动地抓取[万维网](http://baike.baidu.com/item/%E4%B8%87%E7%BB%B4%E7%BD%91" \t "_blank)信息的程序或者脚本。

分布式系统：是许多独立的，[网络连接](https://zh.wikipedia.org/wiki/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%BD%91%E7%BB%9C" \o "计算机网络)的，[通讯](https://zh.wikipedia.org/wiki/%E8%A1%8C%E7%A8%8B%E9%96%93%E9%80%9A%E8%A8%8A" \o "进程间通信)的，并且物理上分离的计算节点的集合。

## 1.4参考资料

1. ……

# 2总体设计

## 2.1需求规定

**1、**功能性需求

**1.1爬虫**

1.1.1.爬虫策略

爬虫策略，应该保证爬虫的下载快速和高效，能解决爬虫面临的反爬虫问题。输入入口URL之后，自动分析网页的组织形态获取新的链接，进行下载。例如输入XX电商主页地址后，自动分析导航菜单，自动分析翻页地址，自动分析详情页的地址等。

1.1.2.URL去重算法

对URL进行去重，已经下载过的，没有进行数据更新的，不再进行下载。去重算法应考虑内存的问题，内存越小越优。

**1.2分布式调度算法**

爬虫任务，可以理解为对一个网站的一次采集过程。

分布式爬虫将所有任务在多台机器上分布式执行（可用多进程模拟）。分布式调度策略，应该将不同网站的URL混合后，分配到多台机器上执行。分布式调度策略的重点在URL的分配策略、失败处理等。

分布式调度应该有多种调度策略，满足不同的场景需求。例如，有的任务必须在特定日期前执行完成，有的任务需要在另一个任务之后执行。

调度算法应该在满足特定的条件下，实现最大的下载量。

**1.3网页自动结构化**

1.3.1.对于电商类网页，能对同一个网站的数据进行自动结构化，生成不同的表，例如商品表、店铺表、评价表等

1.3.2.对于新闻博客类网页，能进行网页正文的自动抽取，对正文进行自动摘要和关键词分析

**2、**非功能性需求

2.1.模块化， 系统中的可变部分，应进行模块化，是可插拔的

2.2.可靠性，分布式系统的每个模块出现异常后，能自动恢复

## 2.2运行环境

开发环境：pycharm

开发语言： Python

服务器操作系统：ubuntu16.04

运行环境： Python2.7

## 2.3基本设计概念和处理流程

1、系统模块

该系统按照用户输入、系统处理、显示输出的流程工作，分为输入模块，预处理模块，分析处理模块，数据存储模块。

1. 输入模块，用户在ＷＥＢ界面输入ｕｒｌ地址，可以输入多个，用回车换行分割，通过数据交换移交给后端
2. 预处理模块，在接收到前端的url后要验证url的合法性，以及是否可链接。
3. 分析处理模块，这里就是主要的系统功能。首先进行查重等必要操作，在请求页面后对页面进行结构分析，数据抓取等工作。
4. 数据存储模块，对数据入库操作

## 2.4尚未解决的问题

无

# 3接口设计

## 3.1用户接口

输入框：用户输入的url地址

软件界面如下：



## 3.2外部接口

需要linux系统支持，redis，mysql支持，多级计算机支持

## 3.3内部接口

系统内部分为预处理、数据抓取、数据分析、数据存储四个子系统。数据抓取内部有分为：request请求模块、查重模块、筛选模块，各模块相互联系。

# 4运行设计

## 4.1运行模块组合

当输入网站ＵＲＬ时，系统首先启动预处理系统，再调用数据抓取系统中的数据请求模块，最后启动数据存储系统，最终保留数据；

## 4.2运行控制

为加快处理速度，此系统支持多线程处理，对于多核心或多路的计算机，勾选启用多线程处理可大大减少处理时间。

用户使用时，使用鼠标键盘或者是触摸屏等终端的设备进行运行时的选择控制。

## 4.3运行时间

系统主要运行时间在网络请求，数据分析，网络请求与网速，端口等实际情况关系密切，本系统采用多线程等多种手段来提高系统运行效率降低系统运行时间，平均每个ＵＲＬ处理时间不超过3s。

# 5系统数据结构设计

## 5.1结构设计要点

本系统中主要使用了scrapy-redis爬虫框架，redis支持，mysql数据库

# 6正文提取算法实现

**6.1**

在 HTML 中,正文和标签总掺杂在一起。不可否认,标签对文字的修饰作用

在词权确定和排序结果上有很大作用。但是,也正因为 HTML 标签和正文互相交

织的复杂和不规范,使得通用的正文抽取变得难以实现,最终不得不针对不同网

站定义不同规则,时空复杂度也大打折扣。

提出此方法核心依据有两点:1、正文区的密度; 2、行块的长度。

依据 1:一个网页的正文区域肯定是文字信息分布最密集的区域之一,这个

区域可能最大但不尽然,比如评论信息较长,或者网页正文新闻较短,而又出现

大篇紧密导航信息时:。

依据 1 和依据 2 相结合,就能很好的实现正文提取。将依据 1 和 2 融合在

行块分布函数里。具体如下:

首先将网页 HTML 去净标签,只留所有正文,同时留下标签去除后的所有空

白位置信息,留下的正文称为 Ctext.

定义 1. 行块:

以 Ctext 中的行号为轴,取其周围 K 行(上下文均可,K<5,这里取 K=3,方向向下, K

称为行块厚度),合起来称为一个行块 Cblock,行块 i 是以 Ctext 中行号 i 为轴的

行块;

定义 2. 行块长度:

一个 Cblock,去掉其中的所有空白符(\n,\r,\t 等)后的字符总数称为该行块的长

度;

定义 3. 行块分布函数:

以 Ctext 每行为轴,共有 LinesNum(Ctext)‐K 个 Cblock,做出以[1, LinesNum(Ctext)‐K]

为横轴,以其各自的行块长度为纵轴的分布函数;

正确的文本区域全都是分布函数图上含有最值且连续的一个区域,这个区域往往含有一个骤升点和一个骤降点。

于是,网页正文抽取问题转化为了求行块分布函数上的骤升骤降两个边界

点,这两个边界点所含的区域包含了当前网页的行块长度最大值并且是连续的。

用数学语言描述就是:

求正文区域所在的起始行块号 X start 和终止行块号 X end ( X 为行号,Y(X)是以

X 为轴的行块长度 ),需要满足以下四个条件:

1、Y(X start ) > Y ( X t )( Y ( X t )是第一个骤升点,骤升点必须超过某一阈值);

2 、Y ( X n ) ≠ 0( n ∈ [ start + 1, start + K ], K 是行块厚度,紧随骤升点的行块长

度不能为0,避免噪声);

3 、Y ( X m ) = 0( m ∈ [ end , end + 1] ,骤降点及其尾随的行块长度为 0 ,保证正文结束 ) ;

4 、 ∃ X , 当取到max ( Y ( X )) 时 , X ∈ [ X start , X end ]( 保证此区域是取到行块最大值的区域)。