# 聚类与多维度可视化

## 十个事件主题

|  |
| --- |
| 主题 |
| MH370 |
| 转基因 |
| 食品安全 |
| 巴黎恐怖袭击 |
| 药品监管 |
| 世界互联网大会 |
| 雾霾 |
| 油价调整 |
| 春运 |
| 二胎政策 |

## 数据源

1. 新浪新闻：

通过对主题在新浪进行搜索，对搜索结果进行爬取。例如：<http://search.sina.com.cn/?q=mh370&c=news&from=channel&col=&range=all&source=&country=&size=&time=&a=&sort=time&t=3_5_6>

对搜索结果爬取的内容有：

|  |
| --- |
| 新闻地址（URL） |
| 新闻标题 |
| 报道媒体 |
| 报道时间 |
| 新闻图片URL |

然后对每条已经爬取的新闻，进入该新闻地址，对该新闻进行进一步的爬取。例如：

<http://news.sina.com.cn/o/2015-10-06/doc-ifximezx1302581.shtml>

对新闻界面爬取的内容有：

|  |
| --- |
| 新闻的整体内容 |

1. 腾讯新闻：

通过主题在腾讯进行搜索，对搜索结果进行爬取。例如：

<https://www.sogou.com/sogou?site=news.qq.com&query=mh370&pid=sogou-wsse-b58ac8403eb9cf17-0004&idx=f&page=1>

由于搜索结果中给予的信息比较少，所以在这个界面上我们只爬取新闻的地址。

|  |
| --- |
| 新闻的URL |

对每条新闻，进入该新闻地址，对该新闻进行爬取。例如：

<http://news.qq.com/a/20151101/034679.htm>

|  |
| --- |
| 新闻地址（URL） |
| 新闻标题 |
| 报道媒体 |
| 报道时间 |
| 新闻评论数（用于代表新闻热度） |
| 新闻整体内容 |

## 爬虫伪代码

在写爬虫程序的时候，利用Python的scrapy爬虫框架。根据以上对数据源的分析，主要有以下几个爬虫：

* **爬虫1（新浪搜索爬虫）**

*开始循环（搜索结果页面的response）：*

*通过XPath规则爬取该界面新闻的所需信息（见数据源），*

*将信息存入数据库表1中*

*将报道媒体存入数据库表3中*

*通过XPath规则得到下一页的URL，爬虫访问并得到response*

*结束循环*

通过这个爬虫，可以得到新浪新闻搜索结果界面上的新闻信息。

* **爬虫2（新浪新闻内容爬虫）**

*对数据库表中的每条新闻URL：*

*爬虫访问该URL，得到response*

*解析response，通过XPath规则得到该新闻的报道内容*

*将新闻内容存入数据库表1中*

通过这个爬虫，可以得到数据库表1中的新浪新闻的新闻内容。

* **爬虫3（腾讯新闻爬虫）**

*开始循环（搜索结果页面的response）：*

*通过XPath规则爬取该界面新闻的URL，*

*对每一条新闻URL，爬虫访问该URL，得到response：*

*通过XPath规则爬取该界面新闻的信息，（除了评论数）*

*将报道媒体存入数据库表3中*

*通过抓包，手动合成储存评论数的文件的URL*

*通过解析json文件，得到评论数*

*将新闻信息存入数据库表2中*

*通过XPath规则得到下一页的response*

*结束循环*

通过这个爬虫，可以得到需要爬取的腾讯新闻的信息。

## 数据库表格

根据之前的分析，数据库总共包含4个表：

* **表1(sinanews)：**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **表格列名** | 新闻ID | 主题  ID | 媒体ID | 新闻URL | 新闻标题 | 报道时间 | 图片URL | 新闻内容 |
| **数据类型** | int | int | int | char | char | char | char | text |

* **表2(tencentnews):**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **表格列名** | 新闻ID | 主题ID | 媒体ID | 新闻URL | 新闻标题 | 报道时间 | 新闻内容 | 评论数 |
| **数据类型** | int | int | int | char | char | char | text | int |

* **表3(media):**

|  |  |  |
| --- | --- | --- |
| **表格列名** | 媒体ID | 报道媒体 |
| **数据类型** | int | char |

* **表4(theme):**

|  |  |  |
| --- | --- | --- |
| **表格列名** | 主题ID | 主题 |
| **数据类型** | int | text |

**数据库的表中，将媒体与主题单独储存在一张表内，便于管理，将主题ID与媒体ID作为该表的主键，在新闻的表中，通过主题ID和媒体ID的外键调用每条新闻的主题与报道媒体。**

## 数据调用

通过PHP对数据库中数据进行调用，在前端进行展示。总共有三个PHP文件。

* **Conn.php**

*PHP连接数据库*

* **sql.php**

*包含conn.php*

*创建一个从数据库取值的类：*

*创建取值函数：*

*连接数据库*

*PHP调用sql语句查询数据*

*返回json编码的数据*

*函数结束*

*类结束*

* **ajax.php**

*包含sql.php*

*如果取值方法method等于取值函数名：*

*调用该函数*

*创建取值函数：*

*实例化sql.php中的取值类*

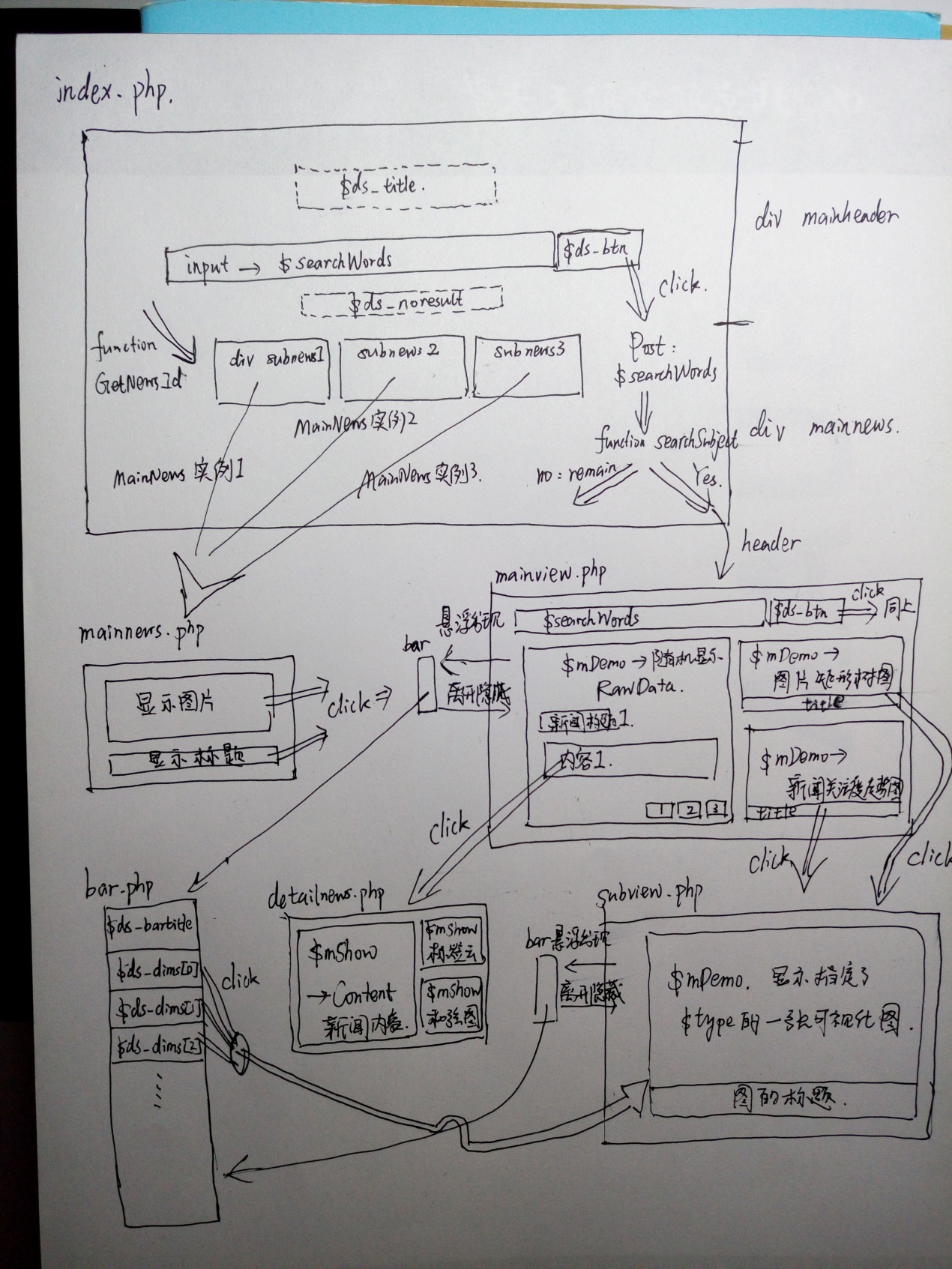
*调用取值类中的取值函数*

*返回json*

*结束函数*

## 前端展示

* **前端设计图片显示**

****

* **前端控制函数**

**config.php**

*说明：存放一些系统变量*

*/\*Mysql数据库参数配置\*/*

$dbhost="localhost";

$dbuser="XXX";

$dbpassword="XXX";

$dbname="XXX";

*\*注:在连接数据库时最好把$conn设成超全局变量，因为很多文件中都会引用*

*/\*项目名和位置\*/*

$config\_name="XXX";

$config\_basedir="http://localhost/XXX";

**design.php**

*说明：存放一些名词，index，mainview和bar将会引用*

$ds\_title = “XX搜索”; *//index上显示的标题*

$ds\_btn = “搜索”; *//搜索按键上显示的内容*

$ds\_noresult = “对不起，没有搜索到相关结果”;

$ds\_bartitle = “XXX”; *//侧边栏的标题*

$ds\_dims = array(); *//侧边栏上显示的不同维度的名称，见DemoTypes*

**stylesheet.css**

*说明：样式表*

**mouseevents.js**

*说明：实现某些效果的js函数*

function showBar(id)

*说明：显示侧边栏*

function hideBar(id)

*说明：隐藏侧边栏*

**index.php**

$latestSubs = array(); *//存储最近三个主题*

$searchWords; *//存储要搜索的关键词*

$mNews = array(); *//存储三个MainNews实例*

function SearchSubject($searchWords)

*说明：访问数据库，由搜索的关键词返回新闻主题的id*

function GetNewsId()

*说明：访问数据库，获得最新的三个新闻主题id，返回id数组*

**mainnews.php**

*说明：在index中被多次引用来显示的几条代表性新闻的结构布局，通过在index中改变$mNews下标来多次显示不同的内容*

**bar.php**

*说明：侧边栏，在mainview.php和subview.php中被引用*

**mainview.php**

*说明：搜索结果的主视图*

$searchWords; *//搜索的关键词*

$subjectId;  *//搜索到的主题id*

$mDemo = new NewsDemo($subjectId);

*//调用get\_content\_addr 得到几种需要显示的代表性可视化图*

**subview.php**

*说明：搜索结果的副视图，点击bar中对应维度后跳转至此，显示bar和一张对应的可视化图*

$subjectId;  *//搜索到的主题id*

$mType;  *//图的type*

$mDemo;*//从mainview.php传递过来*

*//调用get\_content\_addr 得到需要的可视化图*

**detailnews.php**

*说明：点击某条新闻链接后显示该条新闻的详细信息*

$mShow = new NewsDataShow(…);

*//调用*get\_tagcloud(),get\_ chorddiag ()*得到需要显示的可视化图*

**class NewsData**

*说明：存储一条新闻的各个属性*

public:

$url;

$time;

$title;

$subject;

$media;

$pic\_url;

$content;

$remark\_count;

function \_\_construct *//为以上各属性赋初值*

**class NewsDataShow extends NewsData**

*说明：点击一条新闻后单独显示内容的属性与方法*

public:

$tagcloud\_addr; *//获取的标签云位置*

$chorddiag\_addr; *//获取的和弦图位置*

function get\_tagcloud()

*说明：由其父类公共属性获得该条新闻的标签云地址，赋值给$this->*$tagcloud\_addr*并返回之*

function get\_ chorddiag ()

*说明：由其父类公共属性获得该条新闻的和弦图地址，赋值给$this->*$chorddiag\_addr*并返回之*

**class DemoTypes**

*说明：存储可视化展示图的类型，模拟一个枚举结构*

public:

const News = 1; *//最新新闻的漏斗图*

const Pics = 2; *//图片矩形树图*

const Rands = 3; *//随机显示搜索结果*

const Hot = 4; *//新闻的关注度走势*

const Medias = 5; *//媒体柱状图*

const Events = 6; *//事件流淌图*

const Relate = 7; *//新闻相关性*

const Attitudes = 8; *//各新闻的态度*

**class News**

*说明：按主题区分的父类，包含获得该主题下多种数据的方法*

public:

$subject\_id;

function \_\_construct($id)

{

$subject\_id = $5id;

}

Protected:

function get\_data\_by\_url($url)

*说明：访问数据库，由$url获得它对应的一条新闻数据结构并存到一个NewsData实例中并返回之*

function get\_medias()

*说明：访问数据库，由$this->$subject\_id获得由media名为键，该media关于该主题的新闻数为值得数组并返回之*

function get\_urls\_by\_time()

*说明：访问数据库，由$this->$subject\_id获得按时间排序的该主题url数组并返回之*

function get\_rmks\_by\_time()

*说明：访问数据库，由$this->$subject\_id获得按时间排序的该主题remark整型数组并返回之*

function get\_picurls($num)

*说明：访问数据库，由$this->$subject\_id获得按某种排序（默认主键）的前$num个该主题对应的图片url数组并返回之*

function get\_medias()

*说明：访问数据库，由$this->$subject\_id获得由media名为键，该media关于该主题的新闻数为值的数组并返回之*

function get\_titles($num)

*说明：访问数据库，由$this->$subject\_id获得按某种排序（默认主键）的前$num个该主题对应的标题字符串数组并返回之*

**class MainNews extends News**

*说明：用于获得在主页上显示最新的新闻的相关属性*

public:

$title;

$pic\_url;

$head\_addr;

function \_\_construct($id)

{

parent::\_\_construct($id)

get\_title();

get\_picurl();

get\_addr();

}

function get\_title()

*说明：访问数据库，由$this->$subject\_id获得一条代表性的新闻标题，并赋值给$this->$title*

function get\_picurl()

*说明：访问数据库，由$this->$subject\_id获得一条代表性的新闻图片url，并赋值给$this->$pic\_url*

function get\_addr()

*说明：获得该主题的搜索结果页地址，并赋值给$this->$head\_addr*

**class NewsDemo extends News**

*说明：用于获得某主题需要展示的可视化图的方法与属性*

public:

$content\_addr=array();

*//这张图表的html文件所在位置 key: DemoTypes::XXX*

*value:addr*

$titles = array(); *//图表的type对应的标题*

function \_\_construct($myid)

{

parent::\_\_construct($myid);

}

function get\_content\_addr(*$type*)

*说明：根据$type由switch获得不同种类的可视化图并赋值给*$content\_addr并*返回其地址*