

**THU微信资讯平台设计与维护文档**

清华紫荆之声@Fruits Team



2013-12-27

TSinghua University

School of Software

目录

[一、 工程概述 1](#_Toc375992171)

[二、 目录结构说明 2](#_Toc375992172)

[三、 流程结构设计 3](#_Toc375992173)

[3.1 消息传送流程 3](#_Toc375992174)

[3.2 Query处理流程 3](#_Toc375992175)

[四、 校园资讯爬虫 4](#_Toc375992176)

[4.1 爬虫设计的方法 4](#_Toc375992177)

[4.2 缺点分析及维护 4](#_Toc375992178)

[五、 Query的处理 5](#_Toc375992179)

[5.1 判断查询模式 5](#_Toc375992180)

[5.2 分词 5](#_Toc375992181)

[5.3 文字查询处理逻辑 6](#_Toc375992182)

[六、 时间处理 10](#_Toc375992183)

[6.1 类与接口说明 10](#_Toc375992184)

[6.2 系统特性 10](#_Toc375992185)

[6.3 设计实现 10](#_Toc375992186)

[6.4 后续处理 11](#_Toc375992187)

[6.5 缺陷分析与维护 11](#_Toc375992188)

[七、 后台管理 11](#_Toc375992189)

[7.1 设计方法 11](#_Toc375992190)

[7.2 美化样式 12](#_Toc375992191)

[7.3 缺陷分析与维护 12](#_Toc375992192)

[八、 数据来源 12](#_Toc375992193)

[九、 组员分工 12](#_Toc375992194)

[十、 鸣谢 13](#_Toc375992195)

# 工程概述

微信服务号： 清华紫荆之声

开发人员： 王思伦 欧阳方昕 王需 江林楠 洪宇

开发语言： Python 2.7 (请勿使用Python 3.3)

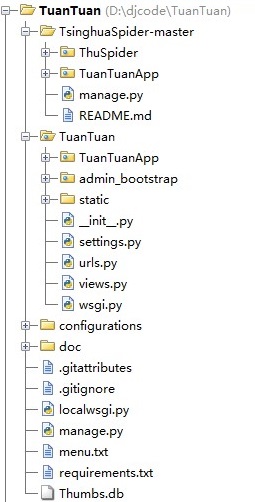
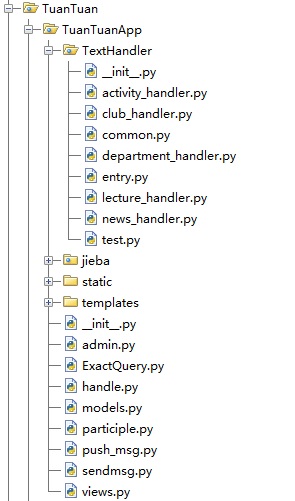
使用框架： Django 1.4+

其他依赖库： HTML解析工具--- BeautifulSoup 4.32

Python分词工具--- jieba

部署方式： 使用nginx + uwsgi (进程数要求≤2)

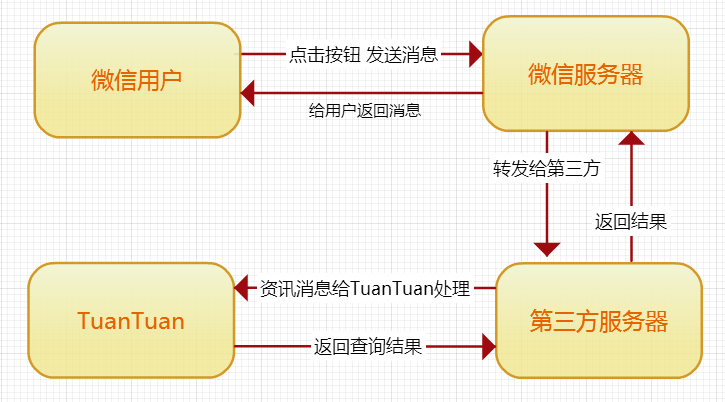
# 目录结构说明

* TsinghuaSpider-master：爬虫程序，主程序在./TuanTuanApp/main.py中
* TuanTuan：Query处理程序目录
* TuanTuan/TuanTuanApp：整个工程的核心部分，用来处理用户的输入，并将结果返回给微信服务器
* TuanTuan/view.py：包含了用户点击图文链接后打开的详情页面的页面处理函数
* TuanTuan/admin.py：控制后台管理页面的显示
* TuanTuan/admin\_bootstrap：控制管理页面的Bootstrap样式
* TuanTuan/model.py：包含数据库文件，利用该文件中定义的类来创建和使用数据库，在\_\_init\_\_.py中定义了用于给用户返回消息的全局变量，当在主页面的view函数中接受到来着用户的输入后，调用位于handle文件中的处理函数，判断用户的输入类型，当确定了用户的具体输入类型后，调用位于send\_msg中的函数进行具体的处理，在函数中通过查询数据库已经修改相应的全局变量值来完成查询，最后更具全局变量将相应结果返回给用户。
* ExactQuery.py：用于判断用户的输入是否为精确输入
* participle.py：将用户的输入进行分词，jieba是我们使用的分词
* TextHandler文件夹中的文件是实现了具体的查询过程，可以根据用户的各种输入返回合理的结果。

# 流程结构设计

## 3.1 消息传送流程



## 3.2 Query处理流程

当TuanTuan接收到Query时，首先判断Query类型属于文本消息(Text)还是事件(Event)：如果是点击事件，则根据按钮ID返回不同结果；如果是文本消息，首先判断此文本能否匹配上一级关键词：“活动” “讲座” “新闻” “任务” “社团” “部门”，如果成功匹配则直接返回相应结果，否则进行二级处理：分析文本的格式，如果符合一定的查询格式如“活动 新年音乐会”则选择精确匹配模式，否则转给模糊查询处理。

# 校园资讯爬虫

## 4.1 爬虫设计的方法

爬虫就像一个默默无闻的搬运工，搜罗清华各处的资讯信息，将其通过微信平台推送给微信用户。爬虫在实现的过程中主要解决了三个问题：

* 1. 怎么获取网络数据？
  2. 怎么分析网络数据？
  3. 怎么保存网络数据？

第一个问题，我们使用python自带的urllib和urllib2库，理论上只要提供给我们链接和账户，我们可以得到任何的数据。将数据从别人的数据中提取出来了，另一个比较重要的问题就是怎么分析你的数据了，这里我们使用了beautiful soup的库来将html格式化，根据不同的网页进行不同的分析，主要包括对演出、新闻、讲座等网页的分析。在beautiful soup中提供了一系列抽取标签的方法，使用起来十分方便，例如：



通过find\_all函数就能找到所有的a标签了。网络的数据格式并一定符合我们需要的，比如原网页是在桌面浏览器中查看的，而我们则是在手机浏览器中查看，这些差异性都需要我们重点考虑怎么提升用户的审美体验，所以在爬虫中，我们也对网络的标签重新整合，美化他的原生页面。Python自带了访问数据库的方法，但这种方法比较麻烦，做为程序员简化我们的思路也很重要，所以我们使用的django的自身封装的api, 通过配置django环境，将我们的应用程序包装成了django应用程序，这样我们只要使用save, create等方法就能直接与数据库联系了。

## 4.2 缺点分析及维护

爬虫程序的特点是对数据提供方依赖较高，无法抽象出比较普遍的方法，所以当数据方的信息出现变化或者格式有些不规范的话就可能引发程序错误（尤其是爬取文化素质讲座时）。目前我们通过爬虫获得两类信息：文化素质讲座和校园演出资讯，具体数据来源见[第八节](#_数据来源)。在后期，需要监视爬虫的表现，发成错误后及时修改。由于数据方有限，对爬虫的拓展性也不是很高，所以如果需要爬取新的信息，只需要在main.py文件中添加一个新的函数，并执行即可。

不过数据不可能无限增长下去，为此我们会自动清空30天前的数据（可任意设置），详见main.py。

# Query的处理

## 5.1 判断查询模式

当接收到用户的输入后，首先需要判断用户进行的精确查询还是模糊查询，我们支持双关键字查询，以空格作为分隔符，判断用户输入的前两个关键字是否存在于下面列表中的行和列中，若存在则存入数组中，判断结束后将去除下列关键字的语句，以及关键字数组传入下层handler逻辑，覆盖如下图所示的查询方式：

## 5.2 分词

#### 5.2.1 方案设计

对于输入串，系统进行分词处理。

设计了分词类Participle，该分词类依赖于Python“结巴”分词库。分词类向用户提供两个分词接口，分别返回精确分词和搜索引擎分词的列表。

使用方法与示例：

from TuanTuan.TuanTuanApp.participle import \*

split = Pariciple(“音乐会”)

words = split.getAccurate() //返回精确分词结果，[“音乐会”]

words = split.All() //返回搜索引擎分词结果[“音乐”, “音乐会”]

在实际的工程中，结合大量的测试，我们从搜索准确度的角度出发，采用了精确分词模式。这样可以过滤掉许多错误的搜索结果。

#### 5.2.2 问题与解决方案

Jieba分词库在程序启动时，会在内存中建立Trie树。这会消耗1至2秒的时间。但是在实际应用时，我们起初将wsgi配置中的并发数设为较高的值。这样用户在输入后会创建一个新进程，直到并发进程数不能增加。因此在进程数创建满之前，对于每一次用户输入，都会创建Trie树，反馈到微信端就表现为超时。

我们经过对各个模块运行时间进行全面测试后发现了问题并且找到了其根源。我们的解决方案是将wsgi的并发数降低，这样就不会创建过多的进程。在实际的使用中，由于我们的算法运行时间非常短，因而在进程数很少的情况下依然可以支持大量的并发。

## 5.3 文字查询处理逻辑

文字查询对外提供的类为TextProcess，其所在的文件为entry.py。该类控制整体的查询逻辑，根据用户的查询，返回包含查询结果的列表。

此外，与TextProcess类相关的文件如下：

1. Participle类：该类负责分词、检索分词结果中的时间。
2. ActivityHandler类：该类负责在活动数据表中查询，返回相应结果。
3. ClubHandler类：该类负责在社团数据表中查询，返回相应结果。
4. DepartmentHandler类：该类负责在部门数据表中查询，返回相应结果。
5. LectureHandler类：该类负责在讲座数据表中查询，返回相应结果。
6. NewsHandler类：该类负责在新闻数据表中查询，返回相应结果。
7. common.py文件：该文件包含上述类中，可以提取出的公共函数，例如filter\_title\_\_contains（根据标题过滤结果）、filter\_keywords（去除分词结果中无用的词汇）
8. test.py文件：测试上述内容的一段代码，支持输入查询文字，返回相应的结果。

假定用户输入的文字为text。

在传给TextProcess类之前，已经由handle.py和ExactQuery.py对text进行过一些处理，并将处理后的text和保存精确查询参数的tags传递给TextProcess的构造函数，然后调用TextProcess类的process方法进行查询。

process方法首先对text进行分词，然后创建相应的Handler（上述2~6）。接下来，检查tags中是否有特定针对“社团、部门、活动、讲座、新闻”的查询，如果有，直接调用相应Handler的handle方法，用result保存其返回值。如果没有，说明需要在每个数据库中进行查询。使用类似责任链模式的处理方法，设定每个Handler的下一个Handler，然后将查询请求的分词结果交给第一个Handler（ClubHandler）。这样，第一个Handler查询到结果之后，将查询结果传送到下一个Handler（DepartmentHandler），下一个Handler再原有结果上继续添加部门信息的查询结果。

获得所有的查询结果之后，回到process方法。该方法将查询结果中的活动、讲座、新闻按照时间进行排序。其中活动、讲座排序的排序方式为：今天-明天-后天-……-昨天-前天，而新闻则按照时间倒序排序。

接下来，按照每个Handler处理结果的mark标识进行从大到小排序。mark标识是描述用户输入与查询信息的匹配程度的。一般来说，与匹配上的字数成正比。

接下来，过滤掉mark较低的查询结果以及与最高mark差距较大的查询结果，从而获得最终的查询结果并返回给调用者。

Tags中除了包含“社团、部门”等关键词之外，还可能包含“标题、地点、时间”这些关键词，这意味着用户希望只在标题、地点或时间中查询。而这部分的处理由每个Handler来进行，因此每个Handler中也保存有相应的tags。其中，具体Handler的查询逻辑如下：

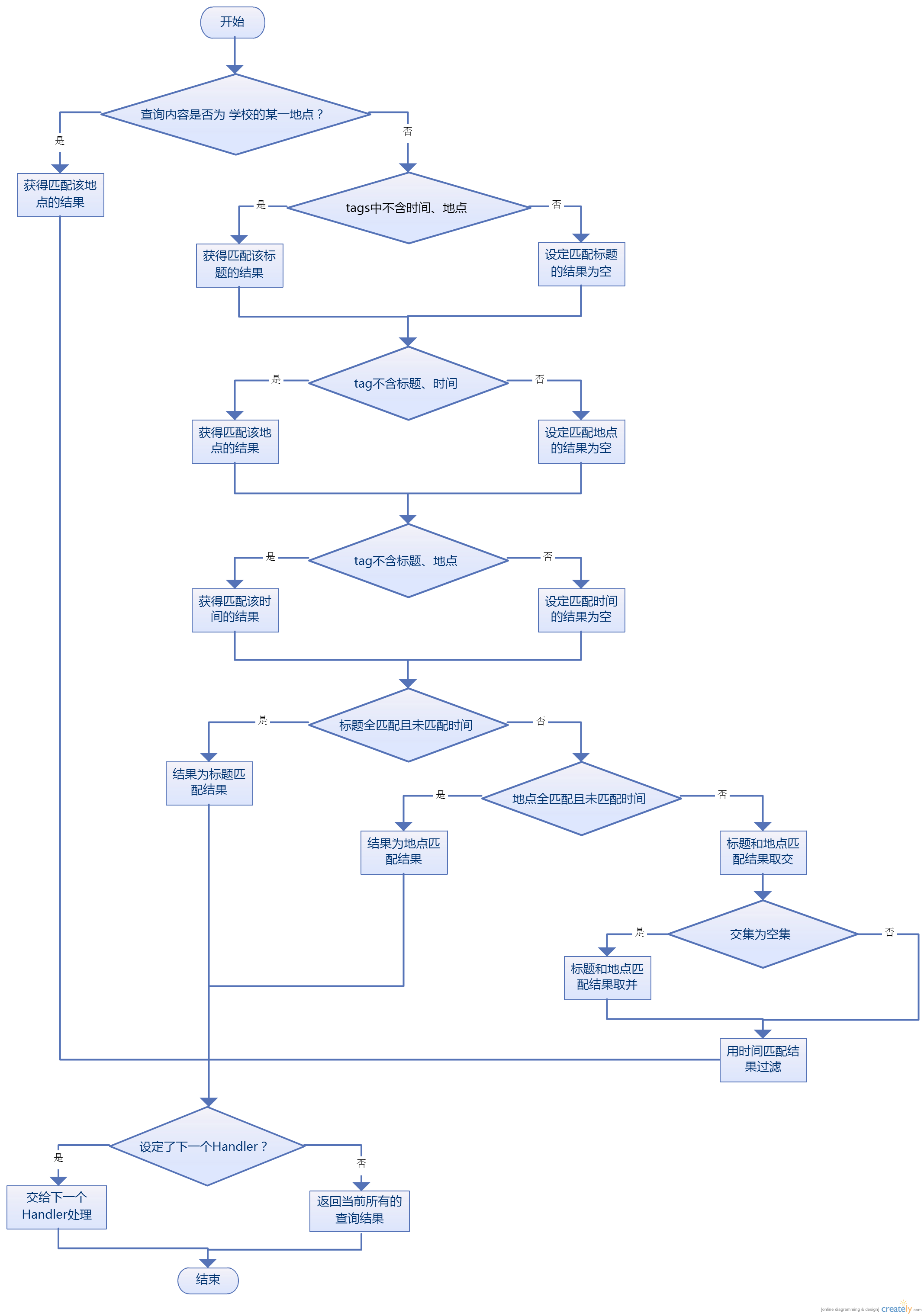


Figure 1-ActivityHandler和LectureHandler处理逻辑

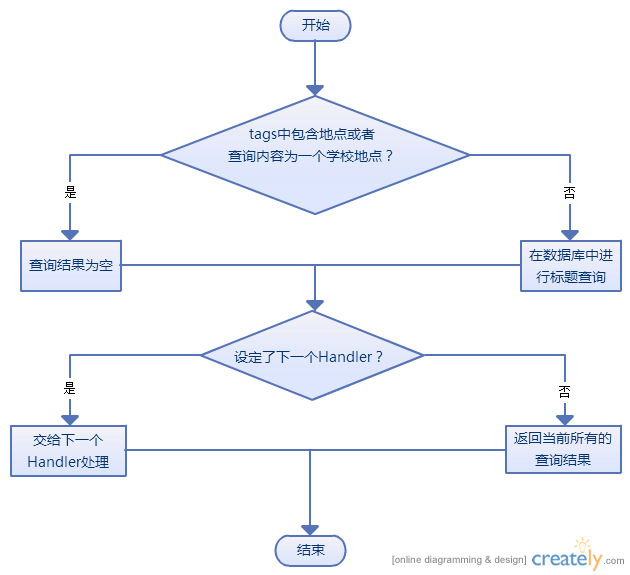


Figure 2-ClubHandler和DepartmentHandler处理逻辑

匹配的规则如下：

假设有效的输入共计n个字符，在某一数据表的title列中，最多能够匹配上m个字符。则返回所有匹配上不少于k个字符的项。k与m的关系为：

|  |  |
| --- | --- |
| K | M |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |
| 5 | 7 |
| M-2, M>=7 | M, M>=7 |

# 时间处理

## 6.1 类与接口说明

查询方式中实现最为复杂的一部分是时间查询。为了对各种可能的时间输入（包括用户的习惯时间用语）进行处理，我们设计了TimeProcess类。

该类接受一个字符串列表进行实例化，对象调用getTimes()接口返回一个起始时间列表，和终止时间列表。字符串列表为一个输入串的分词结果列表。

例：split = Pariciple(“1月4日至1月15日”)

words = split.getAccurate() //分词列表[“1”, “月”, “4”, “日至”, “1”, “月”, “15”, “日”]

timeProcess = TimeProcess(words)

times = timeProcess.getTimes()

处理的返回结果times为[[‘2014-01-04 0:00:00’, ‘2014-01-15 0:00:00’], [‘2014-01-04 23:59:59’, ‘2014-01-15 23:59:59’]]

可以看见，1月4日被处理为2014-01-04 0:00:00和2014-01-04 23:59:59这两个标准时间字符串，它们可以用来直接初始化一个datetime对象。1月15日同理。在后续的逻辑中，这两个时间会被合并为2014-01-04 0:00:00和2014-01-15 23:59:59，这样我们通过输入串得出了用户所期待的时间段的信息。

## 6.2 系统特性

现在系统所支持的输入为诸如2013-01-04 23:00:00“全时间格式”的字串。其中’-’可替换为’年’、’月’、’日’、’号’、’/’，’:’可替换为’时’、’点’、’时’、’秒’，’01’替换为’1’等……以上用户的习惯性输入都是被支持的。

因而用户输入：

1月，4号，2014-1，2014年1-4，19点，19点20分，1月4日，4日19点等各种用户的输入都能智能被系统处理成标准的时间格式。

系统也支持跨年的时间处理，如根据当前时间，“1月”等涉及跨年的时间会被智能地处理。

此外，系统支持“明天”，“后天”，“本个月”，“下个月”等非数值时间输入的识别与格式化，为用户输入提供更加友好与个性化的支持。

## 6.3 设计实现

类的内部所定义的成员方法包括——

**recurTime**:

对输入串进行递归地搜索，从输入串分词列表中提取出时间信息列表。

如“2013年1月清华大学，好开心1-4”被处理为[[‘2013’,’y’,’01’,’m’],[‘01’, ‘-’, ‘04’]

**DateProcess**:

辅助函数。对时间关键词进行识别并插入列表。判定是否继续递归。

**timeFormat**:

对recurTime所得到的分词进行格式化。

具体地对年月日，时分秒信息进行处理，获取“起止”时间。并过滤掉提取出的无效时间。

**extraTime**:

对“明天”“本月”等非数值时间进行识别，并添加到结果列表。

**splitUncommonWords**:

处理分词结果中的“日至”，“日到”等特殊词。使其不干扰时间段的提取。

此外该类中还定义了一些辅助的时间处理函数，但是与程序的逻辑无关。

## 6.4 后续处理

TimeProcess所得到的时间列表并未进行时间的合并与过滤，后续调用timeMerge函数进行处理。调用方式：

time = time\_merge(timeProcess.getTimes())

例：对于时间段输入“1-4至1-15”，经过TimeProcess所得到分别为1月4日和1月15日的起止时间，经过timeMerge后得到的起始时间为2014-01-04 0:00:00，结束时间为2014-01-04 23:59:59。

## 6.5 缺陷分析与维护

暂时不能对“下个月4日”等一类非数值与数值时间的组合进行智能处理，还有待进一步完善。

# 后台管理

## 7.1 设计方法

为了方便后台人员对数据的维护和修改，我们建立了后台管理页面，省去了直接访问MySQL数据库修改数据。在这里我们使用的是Django内置的后台管理页面，只需要修改/TuanTuan/目录下的路由文件urls.py，并在/TuanTuan/TuanTuanApp/中增加admin.py，设定显示的列表、列表项和显示格式。然后就可以直接访问tuantuan.ssast.org/admin/对后台数据进行修改。详细教程请看：Django Book-- [Admin管理界面教程](http://djangobook.py3k.cn/2.0/chapter06/)和[扩展定制管理界面](http://djangobook.py3k.cn/chapter17/)。

## 7.2 美化样式

Django后台管理页面的默认样式很丑，为此我重载了Django后台管理页面的html模板并引入Bootstrap样式。其中Bootstrap样式文件存在于admin\_bootstrap和TuanTuanApp/static中，请勿删除。详细的页面样式修改方法见：[Bootstrap your Django!](https://github.com/SilunWang/admin_bootstrap)

## 7.3 缺陷分析与维护

一般来说校园资讯都有相应的图片链接，目前仅支持输入绝对url，希望后期可以增加“图片上传”功能，建议使用Django的ImageField（预安装PIL库）。

# 数据来源

文化素质讲座：

<http://oars.tsinghua.edu.cn/zzh/30630.nsf/1de?ReadForm&Start=1&Count=50&Expand=2&TemplateType=2&TargetUNID=58F7D30CE9E69CFA482567B800261A8B&AutoFramed>

演出资讯：<http://www.hall.tsinghua.edu.cn/yczx.aspx>

校园新闻：学生清华提供接口

清华人物：<http://news.tsinghua.edu.cn/publish/news/4208/index.html>

社团协会与学校部门：手动输入

注：文化素质讲座的网页格式很不规范，出错可能性很大，需要日后的修缮甚至人工手动维护。学生清华接口使用方法见“学生清华接口API.doc”。目前社团协会与学校部门的资料没有网页版，只能靠人从后台手动输入并调整格式。

# 组员分工

王需：爬虫程序设计 项目部署

18810456160 [xu-wang11@mails.tsinghua.edu.cn](mailto:xu-wang11@mails.tsinghua.edu.cn)

欧阳方昕：第三方服务器与微信之间的通信设计

18810664007 [oyfx11@mails.tsinghua.edu.cn](mailto:oyfx11@mails.tsinghua.edu.cn)

洪宇：Query的模糊查询

13263332086 [hongyu19930808@163.com](mailto:hongyu19930808@163.com)

江林楠：分词逻辑与时间匹配

18810305262 [jln11@mails.tsinghua.edu.cn](mailto:jln11@mails.tsinghua.edu.cn)

王思伦：Django后台管理页面设计与美化 HTML模板设计

18810660040 [badjoker@163.com](mailto:badjoker@163.com)

# 鸣谢

感谢刘强老师在软件工程方面的精彩授课以及对我们自始至终的关怀、刘璘老师在面向对象、抽象用户需求方面给予我们的启发。

同时感谢龚云飞助教和王德希助教在遇到问题时对于我们的帮助，清华大学校团委的支持和赞助。

最后感谢肖贺学长为我们提供的学生清华API、阿里云服务器和需求方面的及时反馈。当然还有所有团队成员，We are Fruits! We are an effective team!