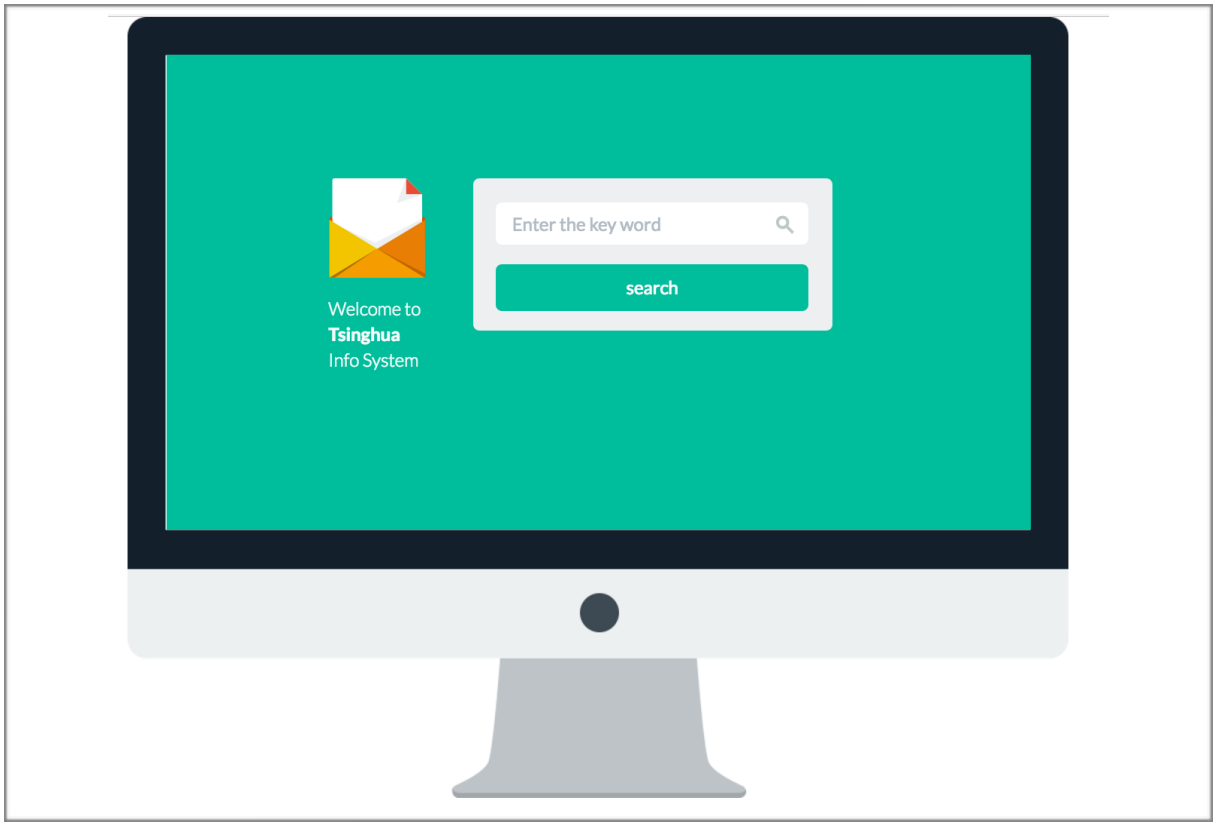
TsinghuaInfoSystem

Software Design Document



张钰晖

计55，计算机系，清华大学

September 10th

2016 夏季学期

1 INTRODUCTION

1.1 Purpose

本篇文档是为清华新闻网信息检索系统TsinghuaInfoSystem编写的设计文档，TsinghuaInfoSystem是清华大学2015-2016学年夏季学期程序设计训练课程的大作业，本文档将具体讨论该系统的设计思路与代码实现，并给出该系统的使用方法，通过阅读此文档，读者将具体了解该系统的实现算法和使用方法，并且了解部分Python软件设计思路。

1.2 Scope

TsinghuaInfoSystem是一个数据量庞大、检索效率高、结果准确度高的信息检索系统，该系统的目标是通过Python的简单爬虫功能，抓取清华网的新闻信息；通过Django搭建服务器，使用Django的MVC框架，实现新闻信息检索系统。下面简述了TsinghuaInfoSystem系统设计的基本要求：

1. 爬取清华新闻网中的新闻信息：根节点：**http://news.tsinghua.edu.cn/publish/thunews/index.html**，从根节点从发爬取所有**news**子域名下的新闻页面，用**Python**包含的工具进行爬取，注意模仿浏览器行为（**UserAgent**）
2. 处理爬取页面信息：为每个新闻页面赋予一个**ID**（便于建立索引），抽取网页关键主体内容（去除标签信息例如**<html>**），包括新闻的标题（副标题）、正文、时间等，对正文和标题进行分词，建立倒排列表（对每个分词）
3. 实现网页，设计**HTML**查询表单(**form**)：文本输入框(**input**)，查询按钮(**button**)，根据输入的查询关键词，返回包含所有关键词的文档，关键词也需要使用**jieba**进行分词

除了以上基本要求外，TsinghuaInfoSystem实现了以下拓展要求：

1. 学习并使用**CSS**框架，美化页面
2. 完善搜索结果显示功能，实现分页
3. 在搜索结果部分对包含关键词的正文进行显示，标红关键词
4. 可以按照时间进行筛选，年份、一周内、一月内等
5. 学习并应用**JQuery**、**AngularJS/VueJS**等框架提升效率

1.3 Overview

本文档将首先讨论该检索系统的设计，具体介绍数据部分的管理和处理方法。接着，本文档将讨论该检索系统的各种实用接口以及检索系统的使用方法。最后，会对该应用进行简单的总结和鸣谢。

2 SYSTEM OVERVIEW

TsinghuaInfoSystem是基于Python 2.7.12进行开发的信息检索系统，开发平台基于OS X 10.11.6。TsinghuaInfoSystem是一个基于网站开发的信息检索系统，提供了原始要求中的所有的功能，并且较为良好的完成了用户交互部分，还提供了更多的扩展功能。

3 SYSTEM ARCHITECTURE

3.1 Architecture Design

应用主要分为爬虫部分、前端部分和后台部分。爬虫部分是指使用基于Python实现的抓取清华新闻网内容，下载内容，处理内容，生成数据库的TsinghuaSpider.py；前端部分主要是指搜索网页的设计、搜索结果的显示等UI的美化；后台部分主要是指基于Django实现的数据传输处理部分，与前端形成交互，后文会具体讨论这一部分的实现。

3.2 Decomposition Description

此部分将按上文所述介绍各个部分的作用，对几个关键的部分进行简单的介绍，分为如下三个部分——Spider、Front End和Back End。如果需要具体查看每个部分的具体使用方法和代码实现，可以查看后文具体介绍。

3.2.1 Spider

Spider部分存放在文件夹根目录下的Spider目录中，包含文件TsinghuaSpider.py。

TsinghuaSpider.py是基于python 2.7.12实现的一个爬虫程序，使用时在对应目录下建立list文件夹，在终端输入“python TsinghuaSpider.py”即可执行该爬虫程序，该程序基于BFS（Breadth-First-Search，宽度优先搜索）思想，爬取清华新闻网的所有新闻。

3.2.2 Front End

Front End部分存放在根目录下的Web/Retrive/templates和Web/static文件夹下。前者包含了两个文件，分别为Retrive.html和Result.html，分别对应前端的搜索欢迎界面和搜索结果展示页面，后者包含了前者所需要的所有JavaScript、CSS外部链接文件。

3.2.3 Back End

Back End部分存放在根目录下的Web文件夹中，基于Django实现，运行时进入该目录，在终端输入”python manage.py run server 0.0.0.0:8000”，将允许通过本地和外部链接的8000端口访问，浏览器地址为ServerIP:8000/retrive。

4 DESIGN

本章将具体讨论上述三个部分的具体实现，同时展示部分笔者认为有意义的思想与代码，供读者参考。

4.1 Spider

TsinghuaSpider.py是基于Python 2.7.12实现的清华新闻获取程序，该程序的主要设计思想是BFS（宽度优先搜索）。

首先建立一个队列（queue）和一个集合（set）用于判重，同时建立四个字典（dict）（dict, file2title, file2time, file2url）分别存放倒序列表（用于信息检索）、文件名对应标题、文件名对应时间、文件名对应链接（用于显示搜索结果，并建立原新闻链接）。

首先在队列里加入初始url(‘/publish/thunew/index.html’),然后每次取出队列里第一个元素，首先判断这个元素是不是清华新闻，不是跳过该论循环（去除css/js/其余网页），同时判断该链接是否已经存在于set（是否已被访问），如果未被访问，把该链

接加入set，同时访问该网站并获取内容，以此链接为基准扩展所有的href，加入队列，同时获取<article></article>的所有新闻主体部分，对该部分进行分词，加入倒排列表（词：文件名），同时记录该文件对应标题、时间、原始链接于对应字典中（使用正则表达式匹配），边爬取边处理文件。

```
class TsinghuaSpider:
      #pseudo code
      def run(self)
            while queue.empty() is False:
                  first = queue.get()
                  if (first[0:17] != '/publish/thunews/'):
                        continue
                  if first not in set:
                        set.add(first)
                        url = rooturl + first
                        try:
                              data = self.get_page(url)
                              hrefs = self.get_href(data)
                              for item in hrefs:
                                    queue.put(item)
                              article = self.get_article(data)
                              if article is not None:
                                    save data to dict
                        except Exception, e:
                              print e
```

4.2 Front End

Front End是指网站的前端部分，主要是指搜索欢迎界面和搜索结果展示界面，为了达到良好的交互，同时符合现代审美，采取Material Design模式（拟物设计），使用了基于Bootstrap开发的Flat UI，同时新闻显示主体部分采用了Django的Python语法，筛选部分（设定按年、月、周、日显示）采用了基于ajax技术的Jquery库。

Django Python语法：

```
    {% for item in items %}

      <div class="row">
        <div class="col-xs-12">
          <a class="lead" href="{{item.url}}">{{item.title}}</a>
          <div class="timelabel"><img src="../../static/img/icons/
svg/clocks.svg" alt="Watches" height="20px"><small>{{item.time}}</
small></div>
          <blockquote>
            {% autoescape off %}
            <p> {{item.text}}</p>
            {% endautoescape %}
          </blockquote>
        </div>
      </div>

      {% endfor %}
```

Jquery样例：

```
 $(document).ready(function(){
    $("#all-button").click(function(){
      $.get("/retrive/", {},
      function(data,status){
        window.open('?key={{key}}')
      });
    });
  });
……
```

4.3 Back End

Back End是指网站的后台部分，主要与前端进行交互，传递需要的信息等。后台基于Django 1.10开发，其中Django采用了MVC框架，在其中建立了App（名为Retrive）后，其中的views.py文件负责处理html的post和request请求，templates文件夹存放了网站的前端主页，views和页面通过render函数传值进行动态交互，减少代码耦合度，便于维护代码。建立页面后，在setting.py中引用App，在urls.py文件中引用该views，同时在其中urlpatterns通过正则表达式匹配的方式跳转到相应网页。

服务器在启动时，会从gl.py中读取数据库并转换成dict存放于内存中，其中在views.py中通过import访问之。

下面讲解一下views.py中各接口的作用与实现：

def get_content(keylist, file)接口会根据关键词列表和文件名，使用正则匹配的方法，获得正文200字并进行关键字标橙。

def get_items(keylist, file)接口会根据关键词列表和文件名，获得该文件的标题、时间、关键词标注的正文、原始链接

def filters(files, arg)接口会根据arg参数，筛选出对应时间的文件名(一年内、一月内、一周内、一天内等)
def retrive(request)是该类的核心，用于获取get的关键字并进行处理，最终返回通过render()函数处理后对应的网页，值得一提的是翻页功能的设计，在全局建立一个当前页数的变量，然后通过get是否有next,prev的请求对变量处理，返回对应的list。

搜索结果会先按关键词出现的次数排序（通过字典映射），再按时间排序，从而按照优先级准确反馈出用户需要的信息。

```
def get_content(keylist, file):
    fileopen = open('./list/' + file)
    content = fileopen.read()
    content = content.decode('utf-8')
    fileopen.close()
    dr = re.compile(r"<p.*?</p>", re.I|re.S|re.M)
    items = re.findall(dr, content)
    content = ""
    for item in items:
        content = content + item
    dr = re.compile(r'<[^>]+>')
    content = dr.sub('',content)
    if len(content) > 200:
        content = content[0: 200] + '...'
    for key in keylist:
        content = content.replace(key, u'<font color="orange">' + key
+ u'</font>')
    return content
```

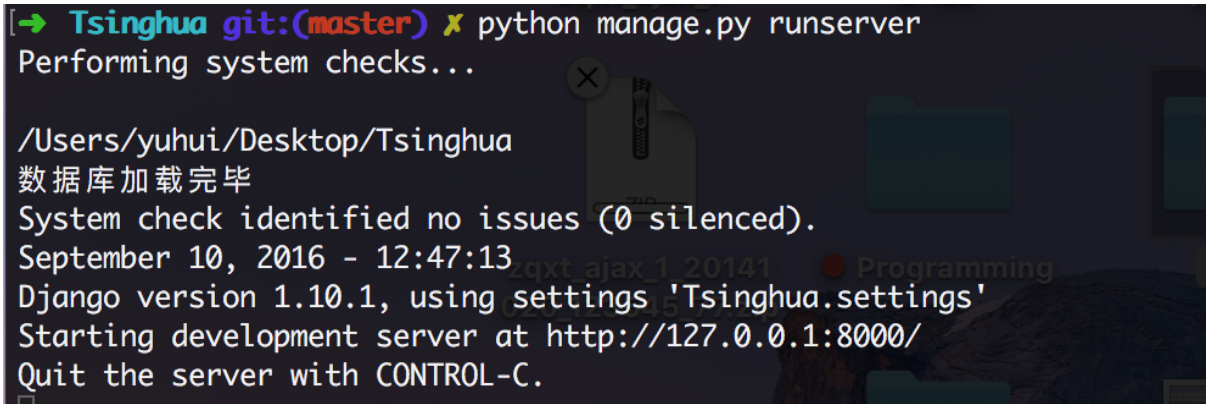
```
def get_items(keylist, files):
    items = []
    if (len(files) != 0):
        for i in range(cur_page * 12, min((cur_page + 1) * 12,
len(files))):
            dict_ = dict()
            dict_['title'] = gl.m_file2title.get(files[i][0])[7: -8]
            dict_['time'] = gl.m_file2time.get(files[i][0])
            dict_['text'] = get_content(keylist, files[i][0])
            dict_['url'] = gl.m_file2url.get(files[i][0])
            items.append(dict_)
    else:
        dict_ = dict()
        dict_['title'] = 'None'
        dict_['time'] = 'None'
        dict_['text'] = 'No result!'
        dict_['url'] = ''
        items.append(dict_)
    return items
```

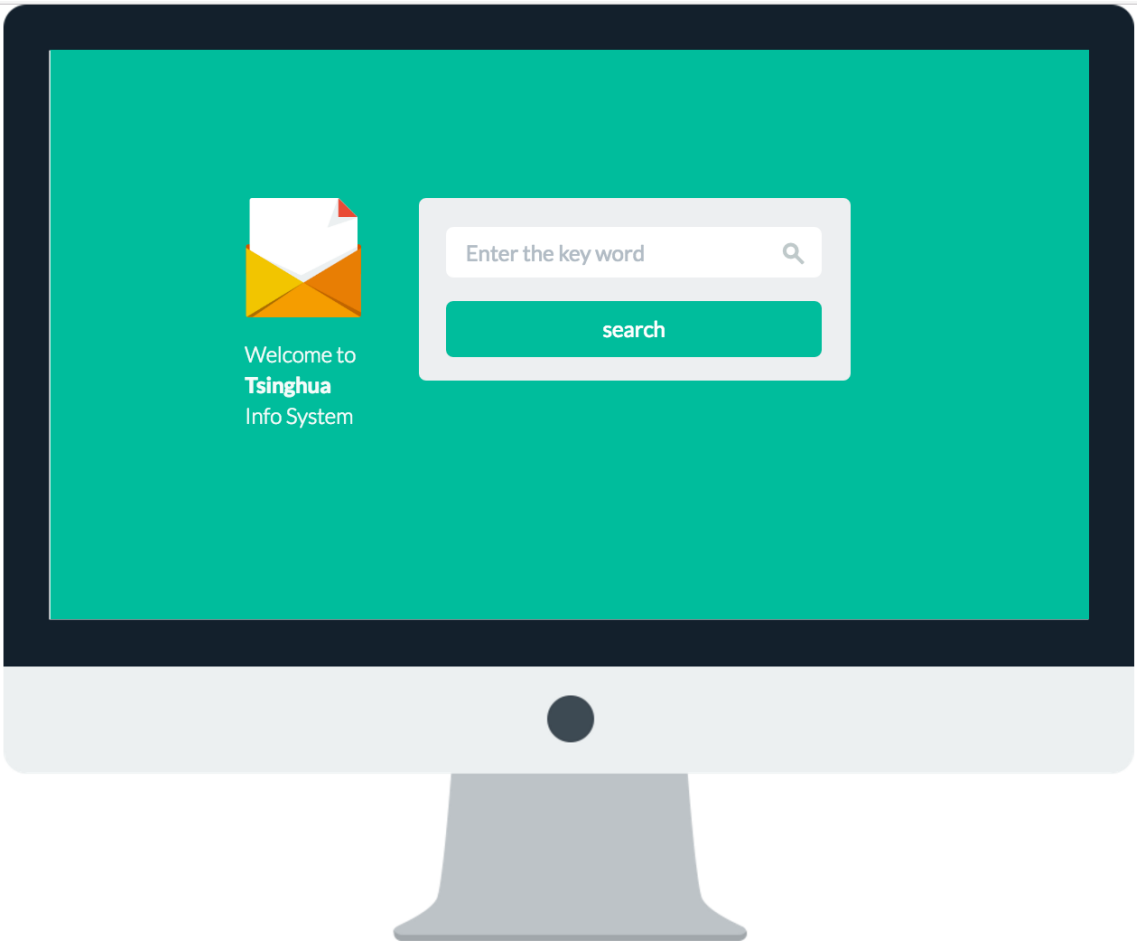
```
def filters(files, arg):
    retfiles = []
    for file in files:
        if (len(file[0]) < 20): continue
        filedate = datetime.date(int(file[0][0:4]), int(file[0]
[4:6]), int(file[0][6:8]))
        datedelte = datetime.date.today() - filedate
        if (arg == 'year' and datedelte < datetime.timedelta(365)):
retfiles.append(file)
        elif (arg == 'month' and datedelte < datetime.timedelta(30)):
retfiles.append(file)
        elif (arg == 'week' and datedelte < datetime.timedelta(7)):
retfiles.append(file)
        elif (arg == 'day' and datedelte < datetime.timedelta(1)):
retfiles.append(file)
        elif (arg ==''): retfiles.append(file)
    return retfiles
```

```
def retrive(request):
    global cur_page
    if request.GET.has_key('key'):
        key = request.GET['key']
        key = key.replace(' ', '')
        keylist_ = jieba.cut(key)
        keylist = []
        for item in keylist_:
            keylist.append(item)

        file2num = dict()
        for item in keylist:
            keyfile = gl.m_dict.get(item)
            if (keyfile == None): continue
            cnt = 0
            for file in keyfile:
                if (file in file2num.keys()):
                    file2num[file] = file2num[file] + 1
                else:
                    file2num[file] = 1
                cnt = cnt + 1
                if (cnt > 5000): break
        file2num = file2num.items()

        if request.GET.has_key('filter'):
            arg = request.GET['filter']
        else:
            arg = ''
        file2num = filters(file2num, arg)
        file2num = sorted(file2num, key = lambda x: x[0],
reverse=True)
        file2num = sorted(file2num, key = lambda x: x[1],
reverse=True)

        if request.GET.has_key('prev'):
            if (cur_page > 0): cur_page -= 1
        elif request.GET.has_key('next'):
            if ((cur_page + 1) * 12 < len(file2num)): cur_page += 1
        else:
            cur_page = 0

        items = get_items(keylist, file2num)

        return render(request, 'result.html', {'key':key,
'items':items,})
    else:
        key = ""
        return render(request, 'retrive.html')
```

5.2 Screen Images

下面是软件使用过程中的截图。






127.0.0.1:8000/retrive/?key=施一公博士&filter=week
Tsinghua Info System
Key words
Search
全部 一天 一周 一月 一年
施一公会见香港浸会大学前校长谢志伟一行
2016年08月30日 11:11:18 清华新闻网
清华新闻网8月30日电 8月27日，清华大学副校长施一公在工字厅会见了来访的清华大学伟伦学术交流中心港方创办人、香港浸会大学前校长谢志伟博士夫妇一行。施一公代表学校对谢志伟一行的来访表示欢迎，对谢志伟长期以来对我校的关心与支持表示感谢，并介绍了学校近期发展，交流了内地高校学科的发展情况。谢志伟回顾了伟伦中心的筹建过程，以及香港与内地开展教育交流以来，内地高等教育发生的巨大变化。施一公（左三）和谢志...
清华大学举行2016级研究生新生开学典礼
2016年08月24日 18:55:34 清华新闻网
清华新闻网8月24日电（记者 李婧 刘蔚如）8月24日上午9点，清华大学2016级研究生新生开学典礼举行。校领导邱勇、陈旭、姜胜耀、史宗恺、邓卫、吉俊民、李一兵、尤政、施一公，及各院系负责人、教师代表等出席典礼，副校长杨斌主持仪式。 在清华大学综合体育馆主会场，包括苏世民书院和全球创新学院首批学生在内的5000余名中外研究生新生参加了典礼，清华大学深圳研究生院和清华-深圳伯克利学院的740余...
清华大学世界文学与文化研究院揭牌
2016年04月13日 10:13:18 清华新闻网
清华新闻网4月13日电 （记者 李含 程曦）4月12日，清华大学外文系成立90周年之际，清华大学世界文学与文化研究院（以下简称“世文院”）正式揭牌。清华大学党委书记陈旭、密歇根大学国际事务副校长詹姆斯·霍洛威(James Holloway)出席揭牌仪式。当天，清华-密歇根博士后研究员学会项目签约仪式举行，清华大学副校长施一公与霍洛威教授共同签署协议。陈旭、霍洛威、郑力、颜海...
新西兰总理在清华畅谈两国的创新与发展
2016年04月20日 14:31:46 清华新闻网
清华新闻网4月20日电 （记者 李含）4月19日，新西兰总理约翰·基(John Key)访问清华大学，并发表题为“创新与发展-新西兰和中国合作伙伴关系”的演讲。校长邱勇在演讲会前与约翰·基总理简短会谈。新西兰驻中国大使麦康年(John McKinnon)、中国驻新西兰大使王鲁彤、新西兰华人议员杨健陪同来访，副校长施一公参加了会谈和演讲会。邱勇与约翰·基总理会谈。记者 张宇 ...
Prev
Next

5 ANALYSIS

本节主要分析该分析平台的性能，由于检索系统将文件中信息直接全部读入内存，故启动服务器需加载数据库，在数据量较大时，这会造成较长的时间。但优点也显而易见，由于python中dict()采用散列与平衡二叉搜索树方式实现，查询复杂度仅O（logn），并且内存中数据访问速度相对于硬盘大大提高，这使得尽管检索系统对数据做了大量的处理，如读取文段，分词，标注关键词，筛选等操作，但查询都可以几乎在常数时间（1s内，甚至远远低于1s）完成，使得具有较高的查询效率。以下给出了数据量n=1000,40000（几乎涵盖了清华新闻网的所有新闻）的数据分析。该分析平台基于OS X 10.11.6

数据量	1000	40000
加载时间	1s内	约120s
查询时间	瞬时	1s内，大部分瞬时
内存占用	约50mb	约1400mb
CPU占用	在一台主机频繁访问时平均低于20%	在一台主机频繁访问时平均低于50%

6 SUMMARY

以上就是TsinghuaInfoSystem全部的设计文档，TsinghuaInfoSystem 是一个高效、准确的信息检索系统。希望能给您的生活带来一些方便。

但由于精力有限，代码实现中可能会有一些不尽如人意的地方，可能会出现一些错误，还请各位使用者多多包涵。同时请包涵设计文档中可能出现的笔误。

在本文的最后，感谢这些天来帮助我的同学，感谢热心网友的帮助文档。