# Week3 体育新闻整合与检索系统

刘泓尊　2018011446　　计 84

## 1 功能展示

### 1.1 数据爬取

#### 1.1.1 新闻数据爬取

本项目的新闻数据库来自于虎扑 NBA 主页（https://nba.hupu.com/），爬取相关新闻数据的工程文件放在/newsspider 中，爬虫框架为 scrapy，具体技术实现请见下文“相关技术”。爬取后得到的数据放在/newsspider/news_data.json 中，总新闻量为 5000+条

#### 1.1.2 球队信息爬取

本项目的新闻数据库来自于虎扑 NBA 主页（https://nba.hupu.com/teams），爬取相关球队数据的工程文件放在/newsspider 中，包括球队名称、创建时间、球员、主教练、主场、所在城市、赛区和球队简介。爬虫框架为 scrapy。具体技术实现请见下文“相关技术”。爬取后得到的数据放在/teamspider/team_data.json 中，爬取范围为 NBA 的 30 支球队。

### 1.2 Web 系统设计及界面美化

本项目 web 系统基于 django 和 html 设计，界面美化采用 css，bootstrap 等技术，同时使用 JavaScript 进行动态页面的设计，扁平化的设计方案使得界面的观感更加贴近用户审美。我为每个界面都加入了“导航栏”，导航栏会随着界面的移动而始终保持在界面顶部，用户可以随时随地点击，回到搜索主页或者球队热度榜。（如图 1 所示）


图 1 每个界面都会带有跟随窗口移动的导航栏

#### 1.2.1 球队主页

球队主页包括元素为：球队名称、创建时间、城市、主场、主教练、球队简

介、球员信息以及相关新闻。（如图 2 所示）相关新闻采用分页展示方式，每页展示 5-6 条新闻。（如图 3 所示）

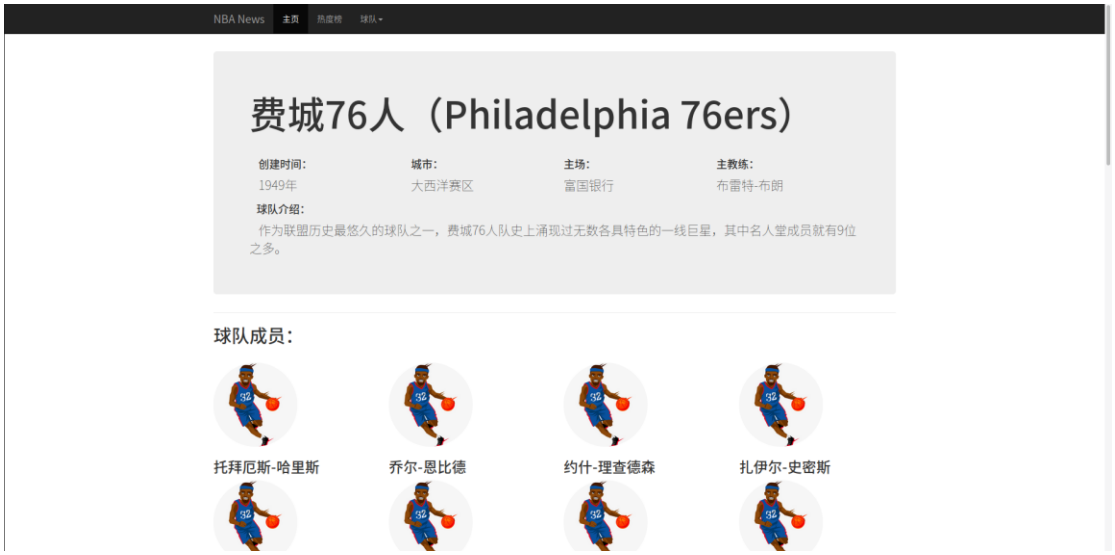


图 2 球队主页打开后的界面


图 3 球队主页相关新闻的翻页功能

### 1.2.2 球队热度榜

球队热度榜通过计算每支球队相关的新闻数量，对对应的球队进行降序排序，以此来获得球队热度排名。（如图 4 所示）界面的表格采用白灰交替的格调设计，使得表格元素更加鲜明整洁。

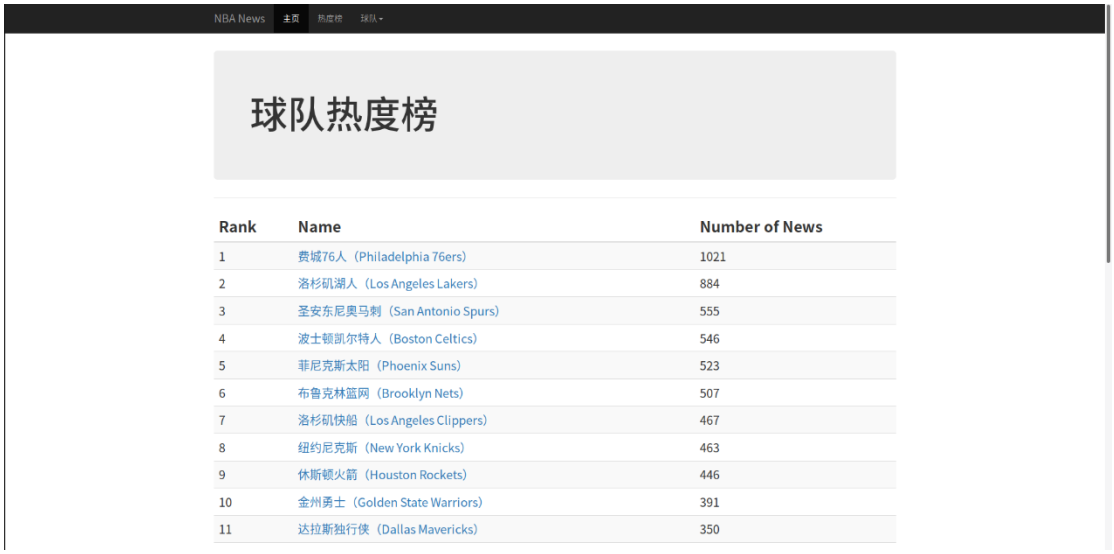


图 4 球队热度榜界面

### 1.2.3 新闻详情页

新闻详情页主要元素为新闻标题、发布时间、来源以及新闻正文，界面右侧列出了新闻热度排名较高的若干球队，同时还有可以到达主页以及球队热度榜的链接。（如图 5 所示）在新闻主体中还会为每个球队名誉球员名建立超链接，点击链接可以跳转至相关球队主页。


图 5 新闻详情页界面

### 1.2.4 关键词搜索（支持高级检索）

我在主页设计了一个搜索框，同时为主页添加了 NBA 相关的背景图和矢量图，搜索框同样采用扁平化设计（如图 6 所示）。在搜索框输入相关信息后，点击“搜索”或按下回车可以显示搜索结果。搜索结果中的关键词可以高亮显示。（如图 7 所示）与关键词相关的新闻会根据其匹配程度排序并显示。新闻数量较多时，可以分页显示，每页显示 10 条。（如图 8 所示）在搜索结果界面显示了搜索条数与搜索所用时间。

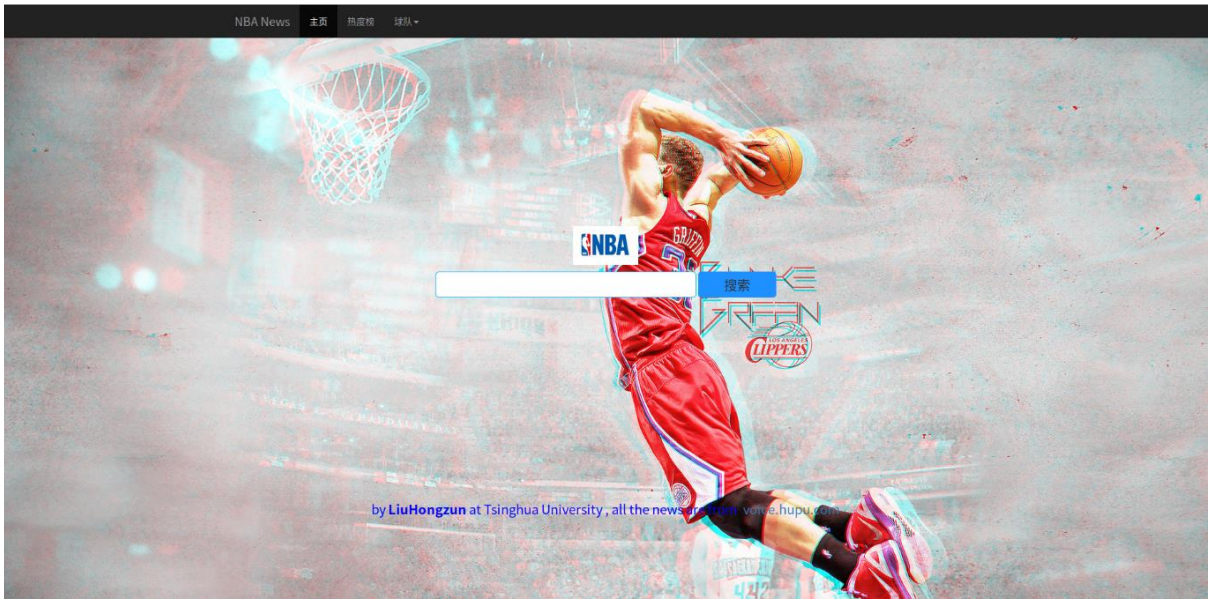


图 6 搜索界面主页

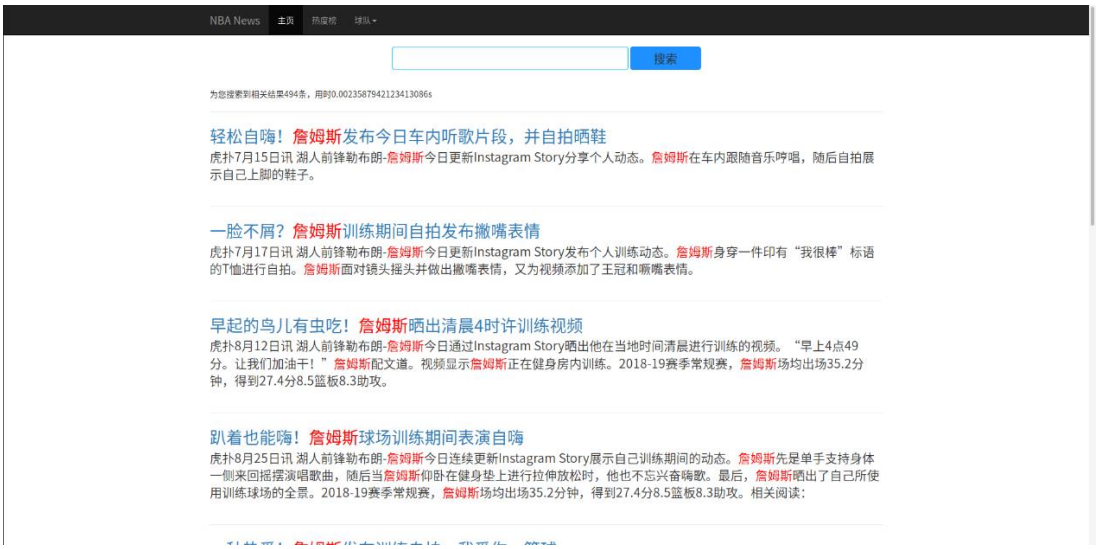


图 7 在搜索界面键入“詹姆斯”后的搜索结果


图 8 搜索结果较多时，分页显示搜索结果

# 2 相关技术

## 2.1 数据爬取

本项目爬虫技术采用 scrapy 库，为每个爬取到的界面解析 HTML，解析采用 xpath+正则表达式。我通过在 teamitems.py 中建立 myitem 类继承 Item 建立爬取数据在内存中的存储，之后通过 myPipeline 继承 pipeline 将内存中的数据导入到 json 文件中。

## 2.2 关键词检索

### 2.2.1 倒排索引与分词

在搜索界面键入关键词之后，关键词传入后端，使用 jieba 进行分词。同时在第一次打开界面时，后端会将新闻数据载入并为新闻中的每个词建立倒排索引，以加速搜索速度。

下表为新闻总量与搜索速度的性能统计信息：

| 新闻总数/条 | 检索时间/s | 平均每 1000 条检索时间/s |
|---|---|---|
| **1000** | 0.0006 | 0.0006 |
| **5000** | 0.0023 | 0.0004 |
| **6000** | 0.0031 | 0.0005 |
| **7000** | 0.0038 | 0.0005 |
| **10000** | 0.0068 | 0.0006 |

表 1 新闻总量与搜索速度的性能统计信息

### 2.2.2 匹配程度

在衡量多关键词的匹配程度时，我首先对数据进行分词，优先查找同时存在两个关键词的新闻。之后使用 tf-idf 技术衡量新闻与关键词之间的匹配程度，对检索到的新闻进行排序。（对含不同数量关键词的新闻分别进行排序，之后再按照含关键词的数量对每一个新闻列表排序），这样就可以使得匹配程度高的词出现在搜索结果靠前的位置。

# 3 感想

本周的学习内容以及作业可以说是这三周以来收获最大的。我初步了解了爬虫技术，Scrapy 框架以及正则表达式的内容，并且对 Web 开发的前后端技术有了初步的了解。虽然刚开始接触这些技术时自己有很强的畏难心理，导致我前两天的进度几乎为 0。但沉下心来仔细分析之后，自己也逐渐熟悉了爬虫和 Web 开发的大致思路，能在短时间内建立起相关框架并产出成果。

第三周之于我最大的收获是，我对于陌生知识的抵触程度大大降低了。这段经历告诉我，今后不管遇到什么陌生的知识，都要迎难而上，沉下心来仔细分析，一切的困难都会被自己的锲而不舍而攻克。

最后感谢老师和助教们的悉心指导，你们的帮助是我平稳度过这三周的极大动力！