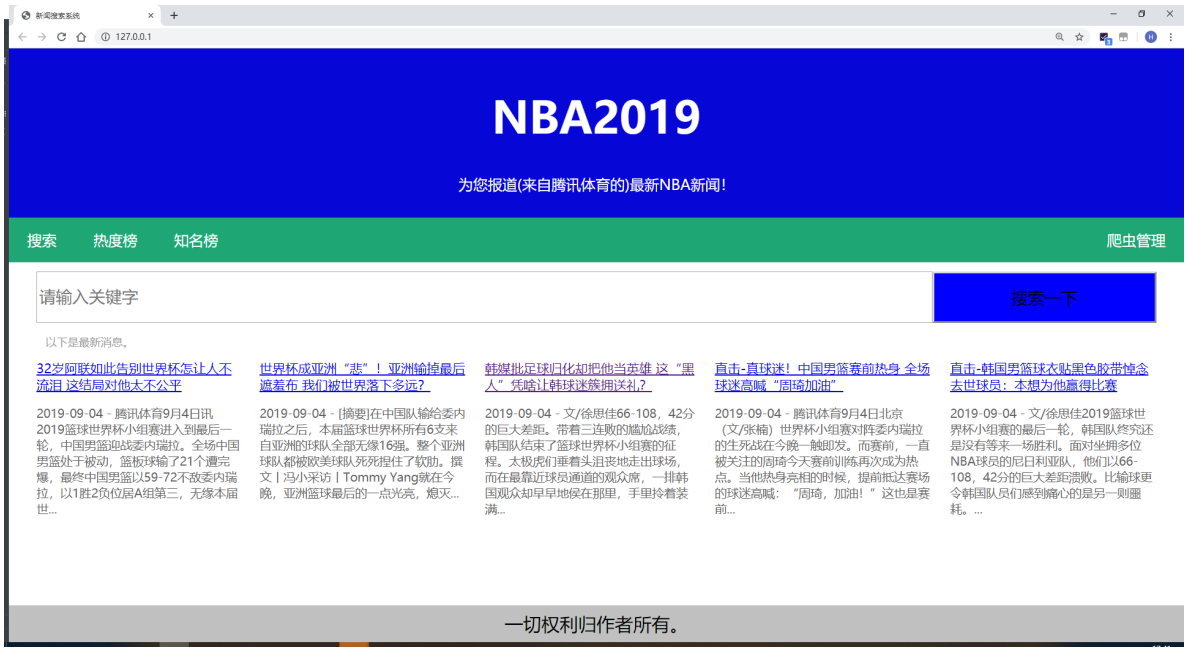
体育新闻整合与检索系统

2017011285 计83 饶淙元

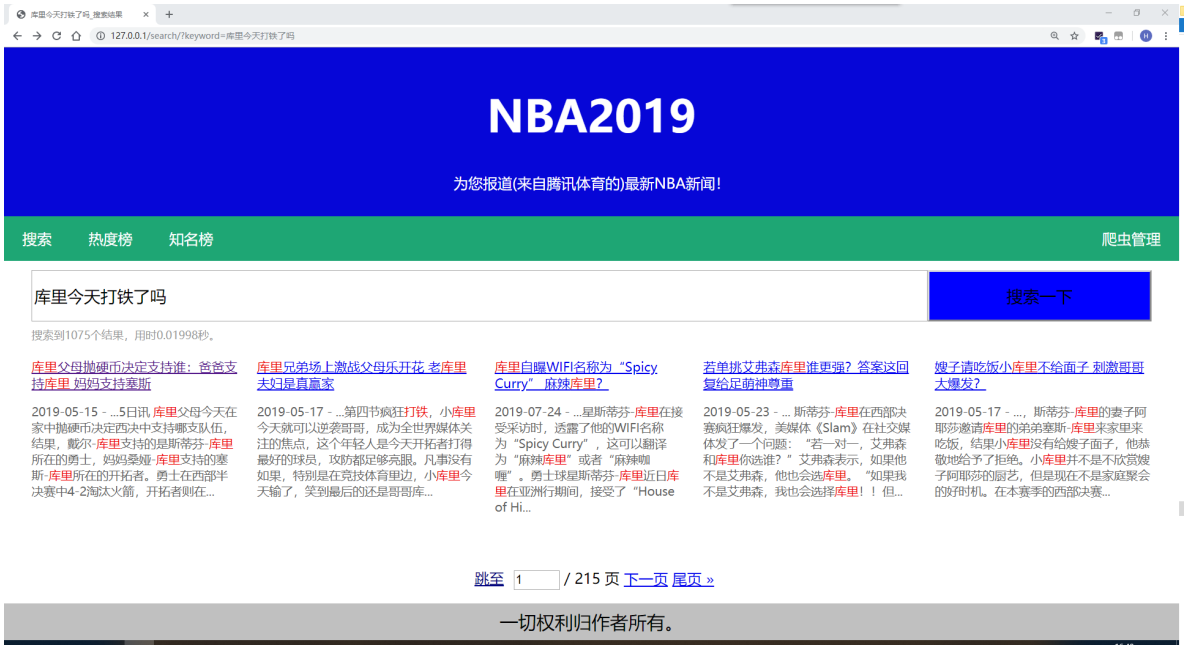
1.功能展示

1.1 主界面



在进入主界面时默认会显示五条最新的消息，此部分没有关键字，只是方便直接查看最近的五条新闻，并且给了各新闻的日期以及前一百字的内容，点击新闻标题可进入新闻查看页。

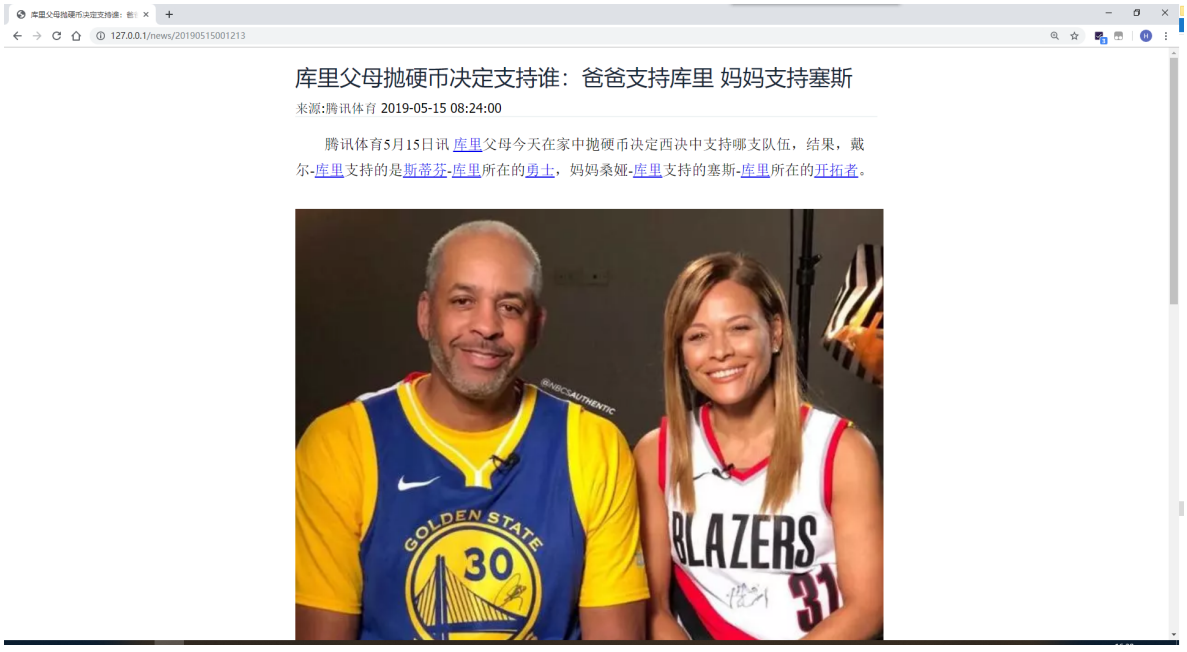
1.2 搜索界面



搜索界面支持多关键字搜索，并且会自动过滤掉停用词，捕获有效信息。如图获取了“库里”和“打铁”两个关键字，“今天”，“了吗”都在停用词表里，因此没有发挥作用。

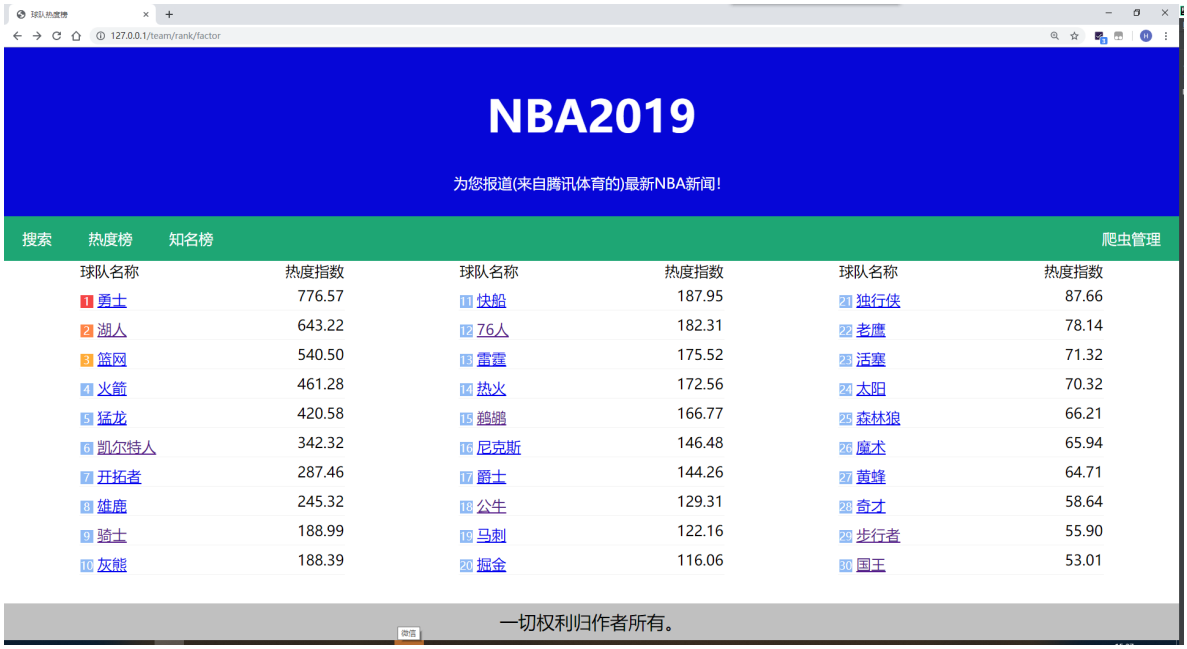
搜索栏下方显示了搜索结果数和搜索时间，新闻显示标题和首次出现关键词部分的段落，对于标题和内容中的关键字给出了加红强调。对于较多结果自动分页，一页最多显示五个新闻预览，如图1075个结果被分成了215页。

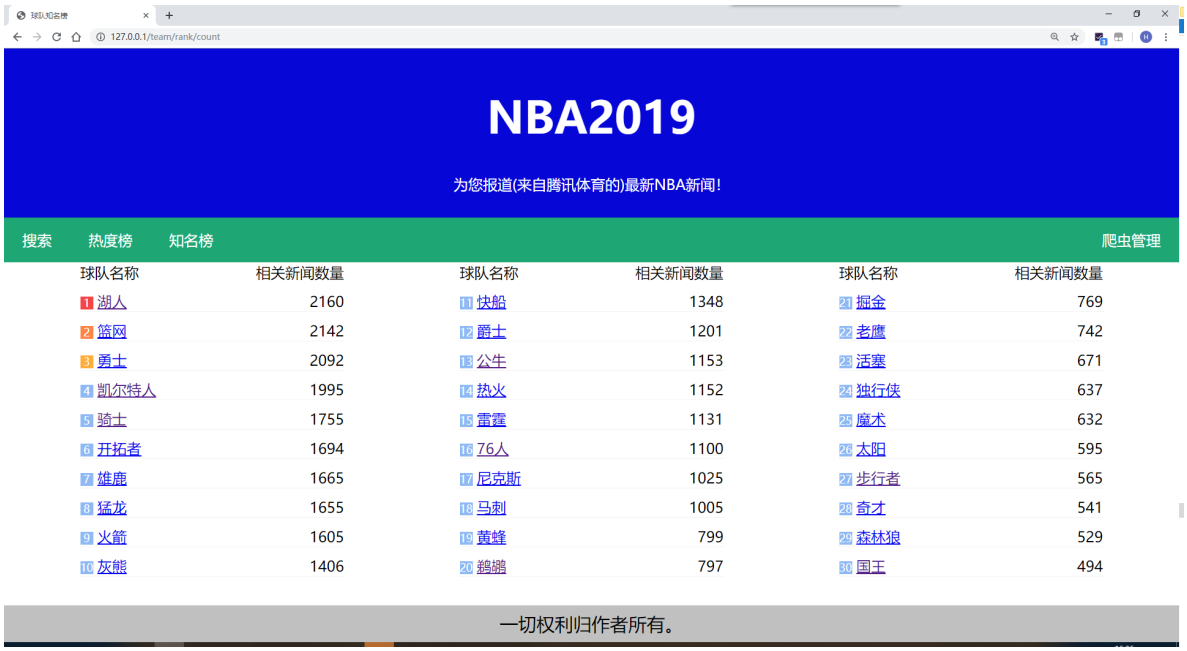
1.3 新闻展示



新闻展示界面采用了和微信公众号类似（雾）的布局，即文字集中在中央的排版。图片穿插在新闻中，队伍名和球员名添加了超链接到队员主页（“詹皇”这样的常见外号也可以识别）。

1.4 球队热榜

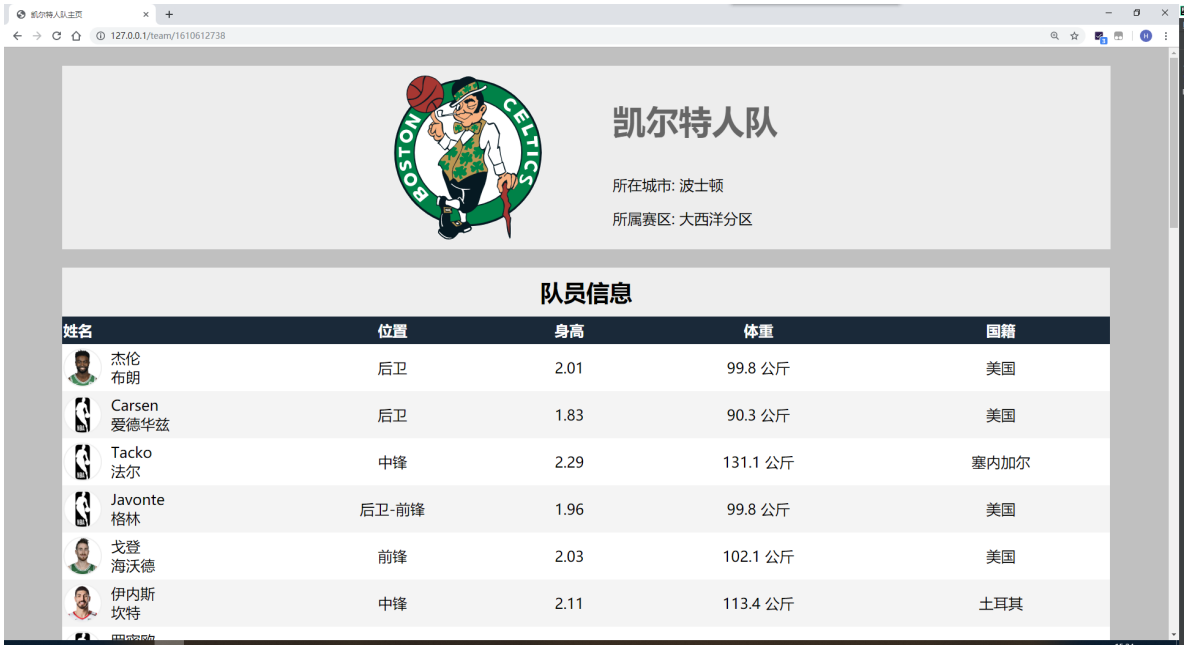




榜单我设计了两个，一个是热度榜，一个是知名榜，在我的定义里热度指一个队伍的成员被新闻提到频率高低，知名是一个队伍的相关新闻数量，比如说在两则新闻里火箭队各被提了一次，其中一条湖人队被提了十次，那么湖人热度高而火箭知名度高。

三十个队伍全部写在一个界面里，分三栏而列，点击队伍名可以进入相应队伍主页。

1.5 队伍介绍



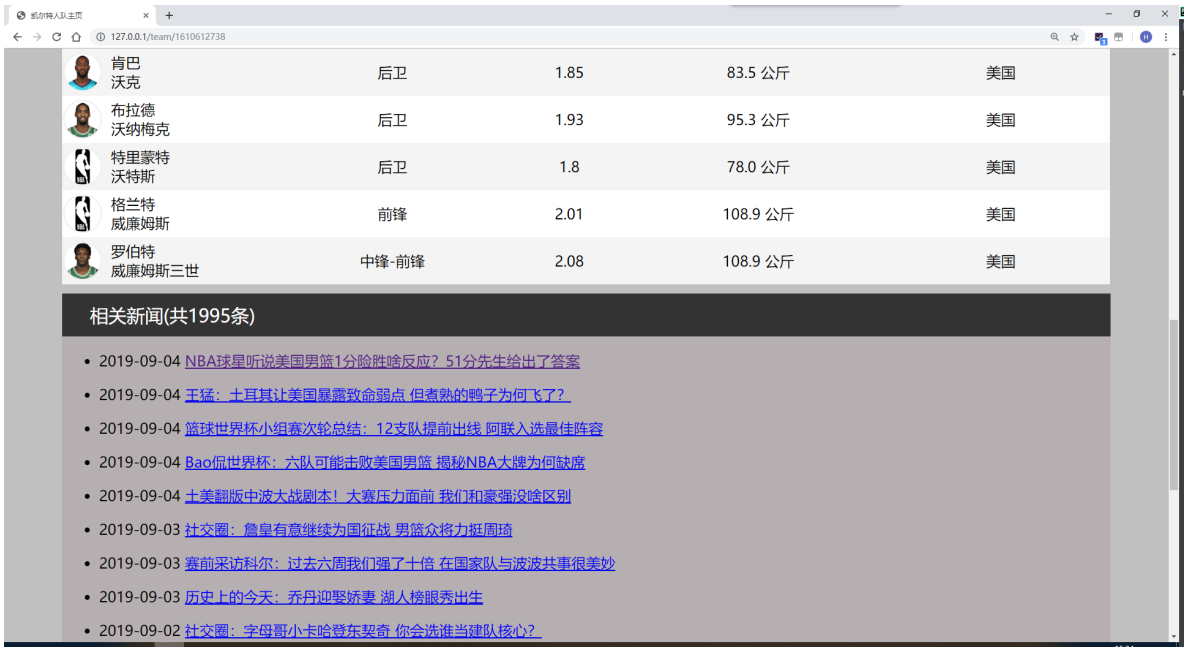
凯尔特人队

所在城市: 波士顿

所属赛区: 大西洋分区

队员信息

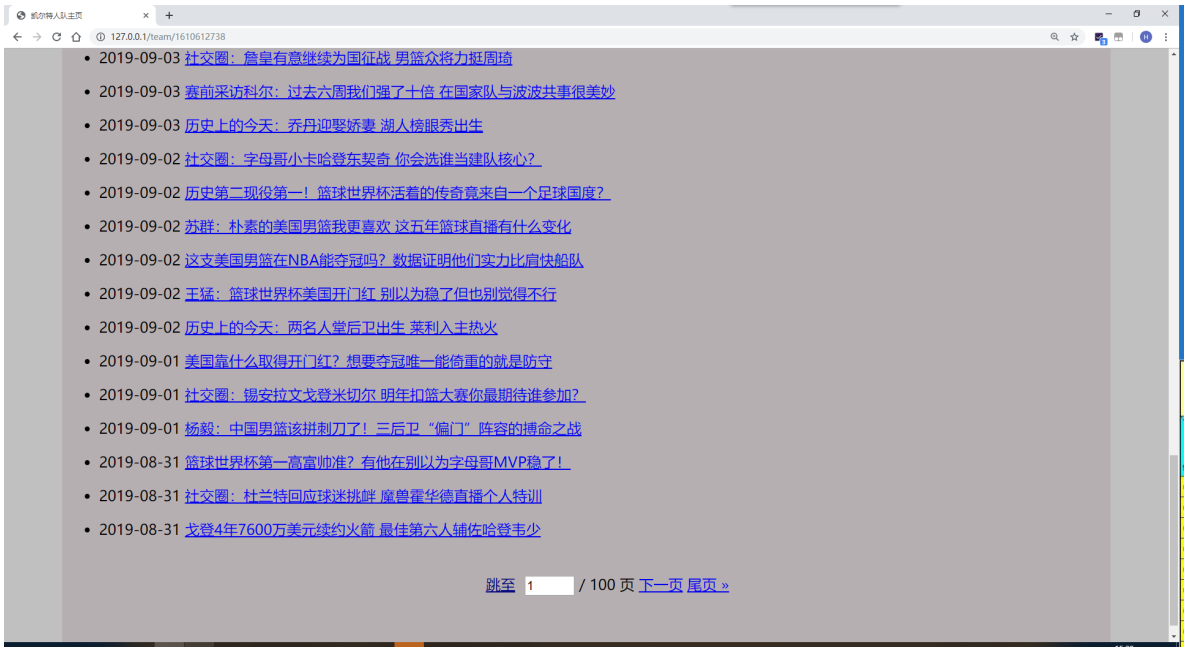
姓名	位置	身高	体重	国籍
杰伦 布朗	后卫	2.01	99.8 公斤	美国
Carsen 爱德华兹	后卫	1.83	90.3 公斤	美国
Tacko 法尔	中锋	2.29	131.1 公斤	塞内加尔
Javonte 格林	后卫-前锋	1.96	99.8 公斤	美国
戈登 海沃德	前锋	2.03	102.1 公斤	美国
伊内斯 坎特	中锋	2.11	113.4 公斤	土耳其



肯巴 沃克	后卫	1.85	83.5 公斤	美国
布拉德 沃纳梅克	后卫	1.93	95.3 公斤	美国
特里蒙特 沃特斯	后卫	1.8	78.0 公斤	美国
格兰特 威廉姆斯	前锋	2.01	108.9 公斤	美国
罗伯特 威廉姆斯三世	中锋-前锋	2.08	108.9 公斤	美国

相关新闻(共1995条)

- 2019-09-04 NBA球星听说美国男篮1分险胜啥反应？51分先生给出了答案
- 2019-09-04 王猛：土耳其让美国暴露致命弱点 但煮熟的鸭子为何飞了？
- 2019-09-04 篮球世界杯小组赛次轮总结：12支队提前出线 阿联入选最佳阵容
- 2019-09-04 Bao侃世界杯：六队可能击败美国男篮 揭秘NBA大牌为何缺席
- 2019-09-04 土美翻版中波大战剧本！大赛压力面前 我们和豪强没啥区别
- 2019-09-03 社交圈：詹皇有意继续为国征战 男篮众将力挺周琦
- 2019-09-03 赛前采访科尔：过去六周我们强了十倍 在国家队与波波共事很美妙
- 2019-09-03 历史上的今天：乔丹迎娶娇妻 湖人榜眼秀出生
- 2019-09-02 社交圈：字母哥小卡哈登东契奇 你会选谁当建队核心？

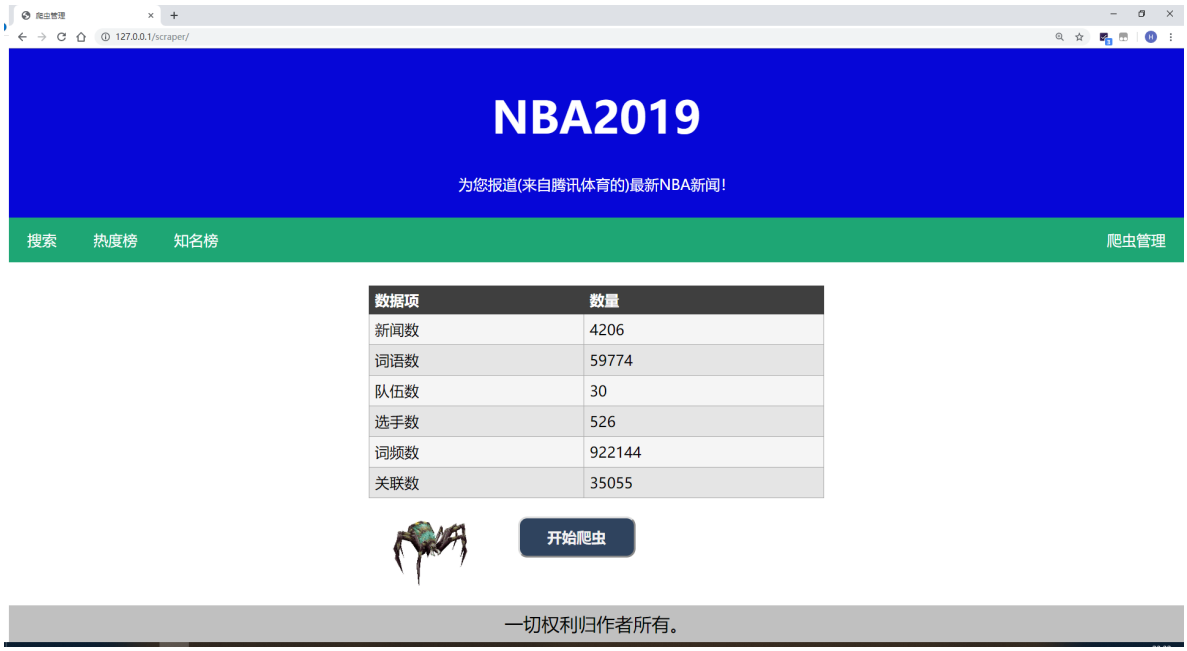


- 2019-09-03 社交圈：詹皇有意继续为国征战 男篮众将力挺周琦
- 2019-09-03 赛前采访科尔：过去六周我们强了十倍 在国家队与波波共事很美妙
- 2019-09-03 历史上的今天：乔丹迎娶娇妻 湖人榜眼秀出生
- 2019-09-02 社交圈：字母哥小卡哈登东契奇 你会选谁当建队核心？
- 2019-09-02 历史第二现役第一！篮球世界杯活着的传奇竟来自一个足球国度？
- 2019-09-02 苏群：朴素的美国男篮我更喜欢 这五年篮球直播有什么变化
- 2019-09-02 这支美国男篮在NBA能夺冠吗？数据证明他们实力比肩快船队
- 2019-09-02 王猛：篮球世界杯美国开门红 别以为稳了但也别觉得不行
- 2019-09-02 历史上的今天：两名人堂后卫出生 莱利入主热火
- 2019-09-01 美国靠什么取得开门红？想要夺冠唯一能倚重的就是防守
- 2019-09-01 社交圈：锡安拉文戈登米切尔 明年扣篮大赛你最期待谁参加？
- 2019-09-01 杨毅：中国男篮该拼刺刀了！三后卫“偏门”阵容的搏命之战
- 2019-08-31 篮球世界杯第一高富帅准？有他在别以为字母哥MVP稳了！
- 2019-08-31 社交圈：杜兰特回应球迷挑衅 魔兽霍华德直播个人特训
- 2019-08-31 戈登4年7600万美元续约火箭 最佳第六人辅佐哈登韦少

跳至 1 / 100 页 下一页 尾页 »

队伍界面分为三个部分，最顶端是队伍log、名字、城市与赛区，中间是一个较长的表格介绍全部队员信息，包括头像、位置、身高、体重、国籍（注：爬自NBA中国，部分头像缺失是该网站数据缺失的缘故），下面是所有的相关新闻，按照从近到远的时间排序，一次显示20条，如图有1995条相关新闻，因此分了100页。

1.6 动态爬虫



如图为爬虫管理界面，展示了目前数据库中的新闻数量、在新闻中被识别到的词语数（即只有搜索时输入这些词语才能搜索到相关新闻）、队伍数、所有队伍选手总数、所有新闻中出现词语的总数

2.性能统计

2.1 新闻数据

由于选定为腾讯新闻的NBA新闻，腾讯滚动新闻里一共给了一百六十多页，八千多条，其中一半是今年的，一半是去年的，因此我只选取了今年的四千余条（吐槽一下大概他们后端有点问题，在第82页到第83页会出现明显时间断层），无法获取更多数据，因此最终结果为四千余条。

需要声明的是，**腾讯新闻的文章长度明显长于虎扑**，腾讯新闻一篇通常在两千五百字左右，而虎扑新闻的字数常在两三百字。我认为对于搜索关键词来说，新闻数据更多的体现在总词数而不是总篇数。

2.2 查询性能

在四千条新闻的情况下，启动服务器后载入jieba词典大概要1~2秒，此后进行搜索时需要时间通常为0.01到0.02秒。

另外由于程序在内存中基本不存东西，所有网页数据都是在加载相应网页时从数据中进行分析（如热榜等），这些也需要时间进行统计，设定上讲知名榜与热度榜是一同加载的，它们一共需要不到0.1秒的时间；加载队伍主页时需要检索相关新闻，需要0.1秒时间；加载首页需要按时间排序新闻然后加载最近五条新闻，需要约0.15秒时间；新闻页面需要获取新闻数据并添加超链接，需要约0.005秒时间。

2.3 后端性能

后端是用户感受不到的效率部分，但也应该属于性能统计的一部分。爬虫爬取大概五分钟四千条，实际动态爬虫工作时是增量更新，爬虫工作时间大概为一秒；导入数据库时一篇新闻用时通常在0.1s左右，动态爬虫如果长时间没更新，需要几秒钟到几十秒钟时间更新数据库，如果短时间没更新，数据库更新时间基本也在1秒以内。

3.技术说明

3.1 动态爬虫

爬虫我使用了上手简洁、爬取速度较快的`scrapy`，没有用到它的pipeline等特性，仅仅是继承自`scrapy.Spider`做了最基础的爬虫类，获取网页信息，然后稍作解析后手动存为`json`。爬取时是从`https://sports.qq.com/l/basket/nba/list20181018164449%s.htm`目录作为根网页，其中`%s`取值是空以及`_2`到`_82`，凭借这82个目录的内容逐条获取各新闻。

最终工作速度较快，首次运行时四千条大概只需要五分钟即可爬完（就是流量跑得有点快）。完成动态爬取时略微遇到一点困难，查询资料发现`scrapy`本身提供了一个进程操作`CrawlerProcess`，但是必须要在主线程中运行，而我计划采用多线程的思路在用户需要动态爬取时创建一个子线程进行爬取，因此遇到问题。最终我直接采用在子线程中调用python的`subprocess`模块运行一个进程，采用最原始的命令行指令控制`scrapy`的运行，完成了爬取操作。

爬取时我会从本地文件检查比对是否已经爬取过目标新闻，只爬前两页未爬取过的内容，这样形成了增量爬取的操作，大大提高了效率，每次爬虫工作时间可能只在一秒左右，剩下的是解析与导入数据库的问题。

完成爬取后在子线程中进行数据库操作——爬取数据与载入数据库我分为两步进行，这样的处理方式缺点是效率可能不及一次完成，但是也不会有明显的效率低下，因为爬虫用的时间只是一秒钟左右；优点是便于管理，确保每次爬取时一定是把新的新闻从目标网站全部爬下来，用户暂停爬虫只是暂停了导入数据库的操作，这样可以有效防止爬取时重复爬取或者缺失等。

3.2 数据库与索引

我使用了django自带的sqlite数据库，属于现学现用，相当多的时间花在了数据库的优化上。

最初对数据库进行操作时，我没有使用bulk操作，也没有用transaction，使用filter时也不会用value等技巧进行加速获取，导致效率非常低下，当时对一篇新闻完成分词并写入数据库需要两分半左右的时间，总的来说需要八天半左右才能完成全部新闻导入，而这显然无法完成大作业要求，因此我连夜爆肝优化。

后来学到了bulk操作，一次关于分词的记录存够999条才写入数据库，如此一来速度大大加快写入速度，一条新闻大约在2秒左右即可完成写入，一共用了不到三个小时完成写入。但我认为这还不够，因为对于动态爬虫而言这个效率还是显得有些慢，并且对于在动态爬取数据时如果一条新闻的数据没有完全写入，这可能对搜索带来一定的干扰，因此我希望有一重缓冲式的操作。

在验收作业前夕，我学会了`transaction.atomic`操作，这样可以让一条新闻的数据在写入时暂时存储在python中，当一条新闻完全处理后自动一次性写入数据库，如此一来减少了局部IO操作，效率又有所增长。另一方面我对Model的设计进行了优化，删除了一些无效信息，例如现在我只关心一条新闻与哪些球队相关，并不关心相关的是哪些词，这样一来记录变得简洁了不少。同时分词策略我也有所改进，所需存储的词频数从三百多万减少到了九十多万。最终一条新闻所用时间大概在0.1~0.3秒，并且随着新词减少新的新闻写入越来越快，只需要半个小时即可完成写入。

最终几经改版，我的数据库模型有以下几个：

WordModel:即所有新闻中涉及到的的词语（jieba词库的子集），其数据项包括词语以及相关的队伍(未必会有)。这里我没有人为指定ID值，由数据库自动分配，实际查询时通常是直接以词语或队伍进行查询，不会涉及到ID值。

NewsModel:新闻信息的存储，包括新闻标题、发布时间、来源、富文本内容、纯文本内容、ID号、相关队伍和包含词语。其中ID作为主键，是直接来自腾讯新闻的URL的最后两部分。

FreqModel:记录新闻与词语关联的模型，它新增了一个权重值来描述关联度。这里同样没有指定ID，实际使用时是用词语或新闻进行筛选。

TeamModel:记录队伍的英文名、中文名、log名、赛区、城市和队伍ID。其中队伍ID作为主键来自NBA中国现有数据，但实际上这并不本质，本来也几乎不会用到它的ID号，一共页只有三十支球队。

PlayerModel:记录一个球员的英文名、中文名、姓、名、队伍中位置、国籍、球员ID、身高、体重和队伍。球员ID作为主键同样来自NBA中国，不过同样不会用到，球员的概念较为薄弱，只是在队伍主页展示所有球员时会涉及到，新闻中的超链接本质上只是词语到球队的映射。

RelationModel:记录新闻与球队关联的模型，它新增了一个关系值来描述关联度。它同样没有显式索引，搜索时都是以球队为关键字进行过滤。

3.3 分词

我采用了jieba库进行分词，早期我采用了搜索引擎模式的分词接口，但是发现效果并不理想，后来为了引入停用词库、减少记录数、为TF-IDF的使用奠定基础，我决定直接采用权重统计，即使用jieba.analyse.extract_tags接口，分析出一段文本中的所有的词语以及它们的权重值，在这之前我设定了IDF和停用词表，最终我不关心一篇新闻中的各个词语出现了多少次，而是关心一篇新闻出现了哪些有效词、权重是多少。

获取到词语与权重后，权重会被记录为FreqModel，此外我事先将各个球队的球员名、外号、球队名存入了WordModel中，一篇新闻中的出现同一个球队的词语时会将权重相加记录，作为这篇新闻和该球队的关系值，为相关新闻和热度榜奠定基础。

3.4 搜索匹配

对于搜索关键字，我先用jieba.analyse.extract_tags进行分词，考察权重，这一过程中会过滤掉停用词（如搜索“我爱篮球”，则“我”和“爱”都是停用词，只有“篮球”是有效词），对于得到的有效词各有一个权重值。然后在数据库中中FreqModel分别匹配各个关键词对应的新闻及权重，将这些新闻的权重分别与关键词权重相乘，然后相同的新闻权重相加，得到新闻搜索相关度。

本质上即将关键词各词语权重与新闻中各词语权重做内积，内积值作为搜索相关度，然后再按相关度从高到低排列。基于前面已经做好的数据库基础，这件事情并不困难，只是做内积我没想到比较好的办法，用的最慢的for循环法，如果能够用到更好的办法或许能够加速不少。