Python 大作业新闻抓取大作业

计 72 陈嘉杰

September 14, 2018

Contents

1 题目说明 **2**

2 功能实现 **2**
2.1 新闻主页 2
2.2 新闻抓取 2
2.2.1 在线接口 2
2.2.2 批量抓取 2
2.2.3 新闻处理 2
2.3 列出所有新闻 2
2.4 新闻详情页 2
2.5 新闻搜索 3

3 程序运行 **3**

4 实现思路 **3**

5 实现细节 **3**
5.1 新闻抓取 3
5.1.1 批量抓取 3
5.1.2 新闻处理 3
5.2 列出所有新闻 4
5.3 新闻详情页 4
5.4 相似新闻 4
5.5 新闻搜索 4

1 题目说明

通过 Python 实现新闻网站的抓取，并通过 Django 框架实现新闻的在线搜索和相关新闻推荐功能。

2 功能实现

2.1 新闻主页

显示一个类似谷歌搜索主页的页面，提供搜索、Feeling Lucky 和所有新闻显示三个按钮。

2.2 新闻抓取

2.2.1 在线接口

在 `/news/scrape` 下提供一个表单，用户可以提交新闻页面，请求新闻的抓取。如果抓取成功，则调转到这个新闻的详情页。

2.2.2 批量抓取

编写了 `scraper.py` ，在腾讯新闻网上，通过三种途径获得新闻的列表，并且对每条新闻进行抓取，保存到数据库中。

2.2.3 新闻处理

下载新闻内容，解析后保存到数据库中，并建立倒排索引。

2.3 列出所有新闻

实现了支持分页功能的全部新闻展示功能，显示每条新闻的源地址、标题、发布时间和摘要。点击标题即可进入该新闻的详情页。数据库中目前已有一万多条新闻的数据。提供了获取新闻时间的显示，一般在 10^{-3}s 的量级。

2.4 新闻详情页

通过编写 `CSS` ，将抓取到的新闻美观地显示，并且在页面底部通过延迟加载的方式，获取相关新闻推荐。

2.5 新闻搜索

输入搜索关键字，后端进行分词后，显示带有这些关键字的新闻。同时支持时间范围的搜索，通过发布时间进行过滤。提供了搜索耗时的显示，一般在 10^{-2}s 的量级，体验良好。

3 程序运行

1. 操作系统：macOS
2. 解释器：CPython 3.7.0
3. 依赖：Django 2.1.1 BeautifulSoup4 4.6.3 requests 2.19.1

4 实现思路

通过 `Django` 的 `ORM` 进行数据的储存，通过 `BeautifulSoup4` 和 `Requests` 进行网站的抓取和解析，通过 `Jinja` 进行静态网页的渲染，通过 `TF-IDF` 和 `Jaccard Index` 进行新闻的搜索和相关新闻的推荐。

5 实现细节

5.1 新闻抓取

5.1.1 批量抓取

通过对腾讯新闻网页的分析，找到它所使用的几处 `XHR` 请求，通过直接请求这些页面，可以获得一个较为格式化的新闻列表。并且通过腾讯新闻的指定日期滚动新闻功能，可以批量抓取指定日期的新闻。为了防止爬虫访问受限制，代码采用了自定义 `HTTP Header` 的方式，并且采用了随机 `User-Agent` 轮换的方法和指数退却（Exponential Backoff）的超时策略。

5.1.2 新闻处理

首先，将网页内容下载下来，使用 `BeautifulSoup4` 进行解析。支持四类已知的新闻页面，它们的主要区别在于，正文、发布时间等采用的 HTML 标签不同，也有的页面是将内容保存在 Javascript 中，代码中都进行了判断和提取，将无关的一些标签去除后，获取完整新闻文本，对此进行分词，并生成摘要。将连接、提取到的标题、正文、发布时间和使用 `jieba` 分词得到的单词存入数据库，建立倒排索引。倒排索引采用 `ManyToManyField` ，即新闻和单词的多对多关系，方便双向的查找。

5.2 列出所有新闻

实现了分页功能，允许指定每页有多少文章，和该页从哪篇文章开始，并在页面底部提供了上一页、下一页的连接。显示每条新闻的源地址、标题、发布时间和摘要，利用 CSS 模仿谷歌的搜索页面，点击标题即可进入该新闻的详情页，点击源连接即可进入新闻的源地址。

5.3 新闻详情页

通过 CSS ，将抓取到的正文进行正常显示，同时提供重新抓取功能，即可以要求后端对该新闻进行重新抓取，方便代码在更新后重新抓取指定页面。页面底部通过 iframe 获取当前新闻的相似新闻，使得在后台进行推荐算法的计算时，用户可以查看新闻全文。

5.4 相似新闻

实现了新闻推荐的在线算法。首先，根据当前新闻的正文，通过 BeautifulSoup4 过滤掉一些标签后获取文本，通过 jieba.analyse.extract_tags 获取 TF-IDF 指数高的词语，查询数据库获取到含有这些词语的新闻，对每条新闻，计算 Jaccard Index 作为这条新闻的权值，最后显示权值最高的三条新闻作为推荐结果。在优化推荐算法搜索时间和搜索结果上，进行了诸多尝试，包括减少数据库查询次数、将一些运算移至数据库中进行等等。

5.5 新闻搜索

对输入进行分词，对每个关键词，通过倒排索引，搜索到相关的新闻，并且根据关键词出现的新闻次数和关键词出现在该新闻的次数求出 TF-IDF 指数，作为权值对新闻进行排序，其中如果某个关键词出现的新闻次数过多，对应的权值会大幅减少，类似于停用词的处理，然后按照分页的请求显示部分结果，并将搜索的关键词进行替换，从而实现高亮显示。经过多次尝试，采用对数计算的 TF-IDF 指数效果较好，并对 IDF 的停用词采用了 0.5 的阈值。