Software Design Document 软件设计文档

新闻搜索引擎 Penguin News Search
魏家栋 2017011445
2018年9月16日

简介

设计目标

该软件希望采用Django框架搭建一个新闻信息检索系统，要求实现：

- 爬取人民网的新闻信息
- 新闻数据的预处理（赋予id、抽取网页关键内容、录入数据库等），对正文和标题进行分词，建立倒排索引。
- 新闻查询主页实现检索功能，实现多关键字的检索功能，检索后跳转到检索结果页面
- 检索结果页面以列表形式展现（如有必要需分页），显示查询结果数量以及查询时间，包含关键字的标题和正文部分予以高亮显示
- 检索结果列表项可以链接到新闻详情页
- 使用css/javascript等对页面布局进行美化

适用范围

适用于2016-2018年人民网新闻的检索。

文档综述

该文档首先对软件进行了一个概述，然后对软件的结构进行了详细分析。接下来介绍了软件的用户界面。最后指出了软件的未来拓展方向。

依赖项、运行和使用

参见README。

版权

软件概述

该软件利用Django框架实现了一个人民网新闻信息检索系统，前端界面类似百度，数据库包含约10000条人民网新闻，查询时间在0.05-0.8秒之间，支持多关键字检索，功能完备。

软件结构

该软件充分利用了Django框架的MTV（Model-Template-View）设计模式，分为3个模块：Model、Template、View，本质上采用了MVC（Model-View-Controller）设计模式。

Model 模块

Model模块负责爬取人民网新闻、建立倒排索引、与数据库进行交互，是整个软件的核心业务逻辑所在。主要包含文件 models.py spider.py invert.py 。 models.py 以Python类的方式定义了数据库所使用的模型，Django可以帮我们将其映射到数据库； spider.py 定义了爬取人民网新闻内容的爬虫，在浏览器地址栏键入"localhost:8000/search/spider"，可进行爬取； invert.py 定义了倒排索引，在浏览器地址栏键入"localhost:8000/search/invert"，可进行倒排索引。
模型主要有2个：News 和 Index。News定义了爬取下来的新闻模型，包括新闻的id、标题、描述、发表时间、正文、原文链接等；Index定义了倒排索引的模型，包括分词产生的关键词、对应的新闻id、共对应多少条新闻等。它们都与数据库相对应。
爬虫通过2种方式进行爬取。一种是利用人民网新闻页面底部和侧栏的“相关新闻”、“热点新闻”、“延伸阅读”等链接，从一个新闻页面跳转到另一个新闻页面进行爬取，这个过程会记录前一个新闻页面上的所有指向其他新闻的链接，因此可以爬取很长时间不中断。另一种是利用人民网的专栏页面，例如“时政”板块的“滚动新闻”专栏，它以列表形式展现，每一项都链接到一个新闻页面，以专栏页面为基础爬取其列表内的所有新闻，爬取速度较快。爬取时会在数据库内进行已爬取新闻的查重，因此不必担心新闻重复的问题。爬虫主要爬取新闻页面中，头部的标题、描述、发表时间等信息，主体的标题、正文等信息。
倒排索引首先利用Jieba对爬取新闻的标题、正文进行分词，将每个词语对应的新闻的id存储在数据库中，建立倒排索引。

Template 模块

Template模块负责前端页面的展示和美化。对应MVC设计模式中的View模块。
主要包含templates文件夹中的html文件和static文件夹中的css文件。html文件利用Django提供的模板机制，可以在模板留下的空位内填充相应的数据，方便了html页面的重用。例如，检索结果页面的布局大致相同，都类似一个列表，但检索结果（网页链接）往往不同，通过Django的模板机制可以很方便地填充对应的检索结果信息。同时，html页面包含了指向css文件的链接，css文件负责html页面的美化。
页面主要有3个：检索主页、检索结果页面、新闻详情页。检索主页和检索结果页面参照百度页面进行设计，检索结果页面利用JavaScript实现了对检索关键词的高亮；新闻详情页参照人民网页面进行设计，主要包含标题、发表时间、正文等，并提供指向原人民网页面的链接。

View 模块

View模块负责将Model模块和Template模块粘合在一起，对应MVC设计模式中的Controller模块。
主要包含文件 views.py 。 类SearchMainView继承generic.TemplateView，控制检索主页的展示；类SearchResultsView继承generic.ListView，控制检索结果页面的展示，将从News数据库查询得到的信息嵌入检索结果页面中，还控制分页、查询内容的分词；类DetailView继承generic.DetailView，控制新闻详情页的展示，将从News数据库查询得到的信息展示在新闻详情页中。
generic.ListView提供了方便的分页机制，只需重写其属性paginate_by即可实现分页功能。

用户界面

在浏览器地址栏键入"localhost:8000/search"，进入检索主页面。主页面类似百度主页面。
在搜索框键入需要搜索的字词，点击"search"按钮，跳转到检索结果页面。检索结果页面以列表形式展现，每条新闻都有标题和描述，检索的关键词会被高亮，类似百度检索结果页面。
在搜索框键入需要搜索的多关键词/句，点击"search"按钮，可以查询到包含其中任意一个关键词的内容。
在检索结果页面点击某条新闻的标题，跳转到新闻详情页。新闻详情页提供新闻的标题、发表时间、正文等，还提供指向原人民网相同新闻的链接。新闻详情页类似人民网的新闻详情页。
在浏览器地址栏键入"localhost:8000/admin"，进入应用管理后台。键入admin的用户名和密码（可以通过

在src目录下运行`python3 ./manage.py createsuperuser`，按照步骤创建admin），登录后可以查看后台的新闻、分词结果等。

未来拓展

由于大作业时间紧迫（只有一周），加上需要学习的东西较多（Python、Django、数据库、html、css、JavaScript等），有许多扩展没有来得及实现。

- 界面的进一步美化。希望能够将页面分为多栏展示，阅读感更为舒适；增加JavaScript动画效果的使用，等等。
- 代码的重构。目前使用的Index模型中，词语存储了其对应的新闻的索引，如果改为与News模型之间的多对多外键可能更好。
- 数据库的优化。目前没有为Index模型的词语建立索引，可能在数据库查询时较慢，也导致倒排索引算法较慢。
- 增强检索功能。增加对新闻发表时间的倒排索引，可以通过新闻发表时间搜索新闻，甚至可以通过关键词和新闻发表时间联合搜索新闻。
- 增加新闻推荐功能。利用文本相似度，在新闻详情页中增加相关新闻的推荐。