PyXinwen 的功能与实现

于是 2017011414

功能概览

PyXinwen 是一个使用 Django 搭建的浏览、检索近期新闻的网站。新闻是从新华网上抓取的，共1987条。

在本地运行服务器之后，输入 `127.0.0.1/news/` 进入。首页显示第1至10条新闻。每条新闻显示其标题、发布时间、摘要，如图：

页面上方的输入框可按关键词、发布日期检索新闻。下方有导航栏。

2018-09-10 07:34:08

“以房养老”是否划算？“以房养老”试点遇冷为何还要在全国开展？“以房养老”能走多远？——日前，银保监会发布了《关于扩大老年人住房反向抵押养老保险开展范围的通知》，将“以房养老”保险由原来的试点城市，扩

药品降价大势所趋 降低抗癌药价还需加把劲

2018-09-10 07:36:18

多省份调降进口药及抗癌药价格，但部分药价降幅不大—— 降低抗癌药价还需加把劲 徐 骏作 （新华社发） 为了让更多患者顺利用上抗癌药物，国家相关部门正不断出台新措施、新政策加快推进进口药及抗癌药降价工作

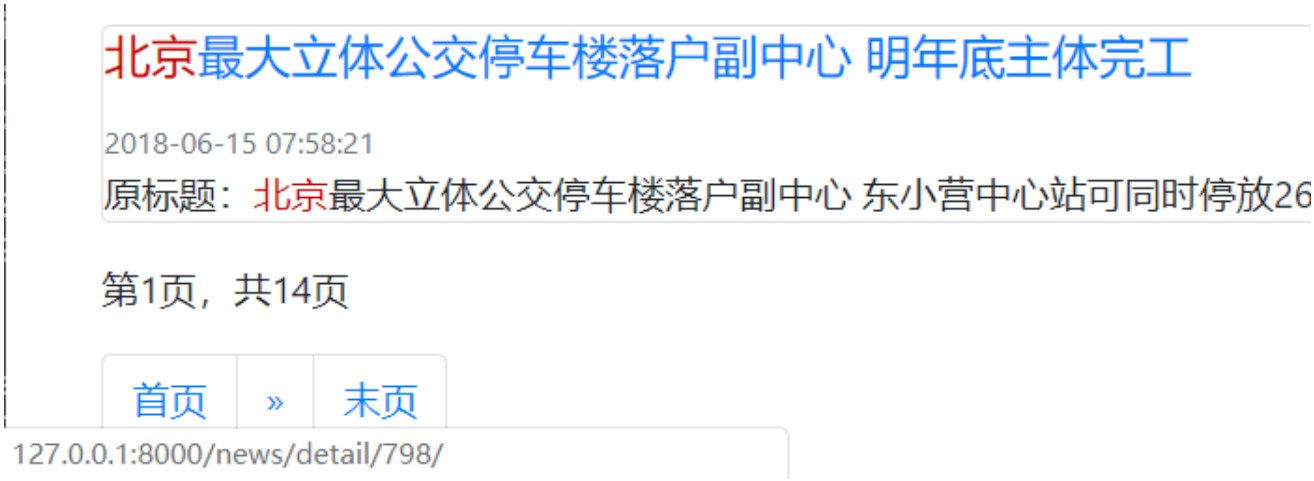
第1页，共199页

首页 » 末页

让我们来检索一条新闻看看。

这是跳转到的检索结果的界面：

上方显示检索出的条数，还有检索耗时。下方呈现检索结果，标题和正文中的关键字会被高亮。页面下方也有导航栏。



点进一条新闻，进入详情页面：

拉到最下方，可以见到相关新闻推荐：

安装了ETC速通卡的车主实现不停车收费又是通过另外一种路径，车主的
设置感知部件，车主进场、出场时感知部件会与车上的电子标签联动得
接连通速通卡实现扣费。据了解，ETC最初应用于高速公路，在经过一定
李洁祎） +1 不停车收费停车场支付宝北京 【纠错】 责任编辑： 聂晨静

北京新建公共停车场须配建充电设施

北京停车场将推广“不停车”收费

北京南站首设网约车专属车位 地铁末班车延长

京津冀高速路出行信息将共享发布

京津冀高速路出行信息将共享发布 三地服务区和收费站将统一标准

这次我们检索”北京 停车“，试一试多关键字检索：

性能信息

共有新闻 1987 条，查询耗时在 ms ~ $10^{-1}ms$ 数量级。

算法简述

查询算法

构建了三个模型，如下：

```
from django.db import models
```

```
class NewsPiece(models.Model):
    news_title = models.CharField(max_length=200)
    news_content = models.CharField(max_length=10000)
    pub_date = models.CharField(max_length=200)
    news_abstract = models.CharField(max_length=200)
    def __str__(self):
        return self.news_title

class Tag(models.Model):
    newspiece = models.ForeignKey(
        NewsPiece,
        on_delete=models.CASCADE,
    )
    name = models.CharField(max_length=255)
    def __str__(self):
        return self.name

class ContentTag(models.Model):
    newspiece = models.ForeignKey(
        NewsPiece,
        on_delete=models.CASCADE,
    )
    name = models.CharField(max_length=255)
    def __str__(self):
        return self.name
```

`NewsPiece` 是新闻模型，`Tag` 是标题中分出的词的模型，`ContentTag` 是正文中分出的词的模型。新闻模型与分词模型是「一对多」的外键关系。

查询时，使用 `NewsPiece.objects.filter(tag__name='关键词')` 以及 `NewsPiece.objects.filter(contenttag__name='关键词')` 进行检索。得到查询结果后，从标题和正文中找到关键词，插入 `<em>` 标签，再传递给模板。

推荐算法

出于尽量模拟真实情况的考量，决定采用「在线」的算法。算法流程如下：

- 点开一篇文章时，对于文章的分词集合中的每一个词，计算其 `tf-idf` 值。`tf-idf` 值是一种综合考量词频、词的独特性的值，值越高表现一个词对该文章的代表性越高。
- 取 `tf-idf` 值最高的两个词，检索包含这两个词的所有文章。
- 对于上步得到的每一篇文章，计算其与本文的 $jaccard$ 相似度。选取相似度最高的5篇显示。

本算法是综合性能和效果考量的。理论上，直接计算全部文章与本文的 $jaccard$ 相似度，便能够得到最精确的推荐，但这样算法耗时过长，极度影响用户体验。因此，在算法中添加了第二步，即依据 `tf-idf` 值先对文章进行筛选，从而减少需要进行 $jaccard$ 相似度计算的文章数量。实测效果较好，是好的折衷算法。

其他说明

`polls` 文件夹是当时按照官方文档进行练习时建立的 App，与本项目无关。

爬虫的代码在 `/news/views.py` 中的 `init_data` 函数中。进入 `127.0.0.1/news/work/` 运行爬虫。