hw5 web搜索引擎

2113644 于洋淼

1. 网页抓取

选取**中国新闻网**，抓取其近5日新闻。 爬取方式如下

1. 首先爬取新闻列表页，爬取所有新闻页的url
2. 爬取所有新闻页，保存对应的标题、正文、新闻种类等信息(保存在json文件中)，并保存html文本为后续构建网页快照做准备
3. 由于中国新闻网的新闻页链接中就包含了新闻种类，因此我们可以很方便获取新闻种类，后续描述用户画像也更加方便

爬取下来的信息保存在文件夹`data/news`中，对应的html文本保存在`data/html`中

核心代码如下:

```
# 获取一页新闻的函数
def get_one_page_news(page_url):

    # 处理响应
    print(page_url)
    response = requests.get(url=page_url, headers=headers)
    status_code = response.status_code
    if status_code == 200:
        html = response.text
    soup = BeautifulSoup(html,"html.parser",from_encoding='gbk')

    news_pool = []
    news_list = soup.find('div', class_ = "content_list")
    items = news_list.find_all('li')

    for i,item in enumerate(items):
        if len(item) == 0:
            continue

        a = item.find('div', class_ = "dd_bt").find('a')
        title = a.string
        url = a.get('href')

        category = ''
        try:
            category = item.find('div', class_ = "dd_lm").find('a').string
        except Exception:
```

```
            continue

        if category == '图片':
            continue

        year = url.split('/')[-3]
        date_time = item.find('div', class_ = "dd_time").string
        date_time = '%s-%s:00'%(year, date_time)

        news_info = [date_time, "http://www.chinanews.com"+url, title]
        news_pool.append(news_info)
    return news_pool

# 爬取新闻的函数
def crawl_news(news_pool, min_body_len):
    i = 1
    for n, news in enumerate(news_pool):
        print('%d/%d'%(n,len(news_pool)))

        req = urllib.request.Request(news[1], headers = headers)
        try:
            response = urllib.request.urlopen(req, timeout=10)
            html = response.read()
        except Exception:
            continue

        soup = BeautifulSoup(html, "html.parser")
        [s.extract() for s in soup('script')]

        try: # 新闻正文
            ps = soup.find('div', class_ = "left_zw").find_all('p')
        except Exception:
            continue

        try: # 新闻标题
            t = soup.find('h1', class_ = "content_left_title")
        except Exception:
            continue
        title = t.string

        try: # 指向其他页面的链接，为后续page_rank做准备
            links = soup.find_all('div', class_ = "intermoren_left")
        except Exception:
            continue

        page_links = []
        for link in links:
            tem = link.find('a')
            link_str = "http://www.chinanews.com"+tem.get('href')
            page_links.append(link_str)

        try: # 获取新闻来源
            news_info = soup.find('div', class_ = "content_left_time").contents[0].strip()
        except Exception:
            continue

        info_str = news_info
        index_of_source = info_str.find("来源：")
```

```
        if index_of_source != -1:
            news_from = info_str[index_of_source + len("来源："):]
        else:
            news_from = '中国新闻网'

        if not news_from: # 如果未注明新闻来源，默认为中国新闻网
            news_from = '中国新闻网'

        body = ''
        for p in ps:
            cur = p.get_text().strip()
            if cur == '':
                continue
            body += '\t' + cur + '\n'
        body = body.replace(" ", "")

        description = body.split('\n', 1)[0].replace("\t", "").replace("\n", "")

        category = re.search(r'http://www\.chinanews\.com/(.*?)/', news[1]).group(1) # 新闻种
类位于链接中
        save_doc(title, news[0], news[1], body, i, page_links, description, news_from,
category)

        with open(f"data/htmls/{i}.html", "wb") as file:
            file.write(html)

        i += 1
        time.sleep(1)
```

2. 文本索引

2.1 预处理

首先需要对文本进行分词，使用`jieba`提供的分词器。

核心代码:

```
analyzer = cut_for_search  # 初始化分词器

def cut_word(doc):
    new_doc = {}
    if doc['title'] is not None:
        new_doc['title'] = " ".join(analyzer(doc['title']))
    else:
        new_doc['title'] = None
    if doc['text'] is not None:
        new_doc['text'] = " ".join(analyzer(doc['text']))
    else:
        new_doc['text'] = None
    if doc['description'] is not None:
        new_doc['description'] = " ".join(analyzer(doc['description']))
    else:
```

```
            new_doc['description'] = None

        new_doc['id'] = doc['id']
        new_doc['date'] = doc['date']
        new_doc['url'] = doc['url']
        new_doc['category'] = doc['category']
        new_doc['page_rank'] = 0
        new_doc['news_from'] = doc['news_from']

        return new_doc
```

2.2 构建索引

对于每个新闻，使用其url作为索引。构建word->url->word frequency的dict，作为倒排索引表

```
# 构建文本索引
index = {}

# 遍历数据框的每一行
for url, row in df.iterrows():
    index[url] = {}

    # 处理标题、描述、文本和新闻来源的文本
    process_text(row['title'], stopwords, index[url])
    process_text(row['description'], stopwords, index[url])
    process_text(row['text'], stopwords, index[url])
    process_text(row['news_from'], stopwords, index[url])

    # 删除空字符串
    if index[url].get('') is not None:
        del index[url]['']

# 构建倒排索引
inverted_index = {}
for url, words in index.items():
    for word, frequency in words.items():
        if word not in stopwords:
            inverted_index.setdefault(word, {}).update({url: frequency})
```

2.3 tf-idf

使用tf-idf来衡量单词权重，为了避免文档长度影响判断，tf的计算方式没有选择单词占文档的比重，而是只使用了单词出现次数。tf-idf的计算公式如下:

1. **TF**

$$\mathrm{TF}(t, d) = \log_{10}(\text{词t在文档d出现的次数}+1)$$

其中，t 是指定的词，d 是文档。

2. **IDF**

$$\text{IDF}(t, D) = \log\left(\frac{\text{语料库D的文档总数}}{\text{包含词t的文档数}}\right)$$

其中，t 是指定的词，D 是语料库。

3. **TF-IDF：**

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

```
# 计算词频
word_frequency = {}
for url, words in index.items():
    for word, frequency in words.items():
        word_frequency[word] = word_frequency.get(word, 0) + 1

# 计算逆文档频率
word_idf = {}
for url, frequency_dict in index.items():
    for word, frequency in frequency_dict.items():
        word_idf[word] = math.log(len(index) / frequency)

# 计算tf
tf = {}
for url, words in index.items():
    tf[url] = {word: words[word] for word in words}

# 计算tf-idf
tf_idf = {}
for url, words in index.items():
    tf_idf[url] = {word: frequency * word_idf[word] for word, frequency in words.items()}
```

3. 链接分析

根据每个网页保存的page_link，计算每个网页的page_rank

```
# 计算 PageRank
digraph = nx.DiGraph()
digraph.add_nodes_from(url_total_list)
for url, url_list in url_list_disk.items():
    for _url in url_list:
        if _url in url_total_list:
            digraph.add_edge(url, _url)

result = nx.pagerank(digraph, alpha=0.85)
```

4. 查询服务

除了基础搜索，提供站内查询、短语查询、通配查询、查询日志、网页快照、日期限制、标题搜索等高级功能

4.1 基础搜索

基础搜索提供站内查询功能，搜索步骤为

1. 对输入的关键词及历史查询记录分词(历史记录的使用是为了实现个性化查询功能)
2. 判断搜索模式(全文or标题)
3. 计算每篇文章、关键词、历史记录的tf-idf
4. 计算余弦相似度
5. 根据余弦相似度和page_rank给文档排名

```
# 基础搜索
def main(input_word: str, history_words: list, is_title_only: bool = False) -> list[tuple[str,
float]]:

    # 对输入的关键词进行分词
    split_input = list(cut_for_search(input_word))
    split_input.sort()
    if '' in split_input:
        split_input.remove('')
    if ' ' in split_input:
        split_input.remove(' ')

    # 对历史记录进行分词
    split_history = []
    for i in range(len(history_words)):
        temp_split = list(cut_for_search(history_words[i]))
        for i in temp_split:
            if i in ['', ' ']:
                pass
            elif i not in split_history:
                split_history.append(i)

    # 判断搜索模式，全文搜索或标题搜索
    if not is_title_only:
        tf_dict = tf
        idfs = idf
        word_sets = word_set
    else:
        tf_dict = tf_title_only
        idfs = idf_title_only
        word_sets = word_set_title_only

    tfidf_dict = {}  # 存储每一篇文档的向量(tf-idf)
    for k, v in tf_dict.items():
        tfidf_dict[k] = computeTFIDF(v, idfs)

    key_tfidf_dict = {}  # 存储关键词的tfidf。筛选出tf-idf最大的前key_valid_number个词，降序排列
    for k, v in tfidf_dict.items():
        key_tfidf_dict[k] = sorted(tfidf_dict[k].items(), key=lambda d: d[1], reverse=True)
[:key_valid_number]  # d.items() 以列表的形式返回可遍历的元组数组
    key_tfidf_list = list(key_tfidf_dict.values())  # 将结果转化为list
    key_tfidf_url_list = list(key_tfidf_dict.keys())
    len_key_tfidf_url_list = len(key_tfidf_url_list)

    tf_input = computeTF(word_sets, split_input)  # 查询的tf
    tfidf_input = computeTFIDF(tf_input, idfs)  # 查询的tf-idf
    key_input = sorted(tfidf_input.items(), key=lambda d: d[1], reverse=True)
[:key_valid_number]  # 查询的前100个关键词
```

```
    len_key_input = length(key_input)

    if len_key_input == 0:
        raise KeyError  # 输入关键词和历史记录都为无法匹配到关键词，则返回错误

    tf_history = computeTF(word_sets, split_history)  # 历史记录的tf
    tfidf_history = computeTFIDF(tf_history, idfs)  # 历史记录的tf-idf
    key_history = sorted(tfidf_history.items(), key=lambda d: d[1], reverse=True)
[:key_valid_number]  # 历史记录的前100个关键词
    len_key_history = length(key_history)

    # 余弦相似度计算
    key_results = []
    key_results_index: list[int] = []  # 用于存储index，方便history_words的调用
    for i in range(len_key_tfidf_url_list):  # 遍历每个文档
        num = 0
        _key_tfidf_list = key_tfidf_list[i]
        for _key_input in key_input:  # 遍历每个关键输入词
            if _key_input[1] != 0:
                for __key_tfidf_list in _key_tfidf_list:  # 遍历每个文档内的每个关键词
                    if _key_input[0] == __key_tfidf_list[0]:  # 若为相同单词
                        num = num + _key_input[1] * __key_tfidf_list[1]
        cos = round(num / (len_key_input * length(_key_tfidf_list)), 4)
        key_results.append((key_tfidf_url_list[i], cos))  # 存储第i个文档的余弦相似度
        if cos > 0:
            key_results_index.append(i)
    if len(history_words) > 0:  # 没有历史记录时不计算历史记录的相似度
        history_results_dict = {}
        for i in key_results_index:  # 遍历每个文档
            num = 0
            _key_tfidf_list = key_tfidf_list[i]
            for _key_history in key_history:  # 遍历每个关键输入词
                if _key_history[1] != 0:
                    for __key_tfidf_list in _key_tfidf_list:  # 遍历每个文档内的每个关键词
                        if _key_history[0] == __key_tfidf_list[0]:  # 若为相同单词
                            num = num + _key_history[1] * __key_tfidf_list[1]
            history_results_dict[i] = ((key_tfidf_url_list[i], (round(num / (len_key_history *
length(_key_tfidf_list)), 4))))  # 存储第i个文档的余弦相似度

        results = []
        for i in range(len_key_tfidf_url_list):
            if key_results[i][1] == 0:
                pass
            elif j := history_results_dict.get(i):
                results.append((key_results[i][0], key_results[i][1] + j[1] / 10))  # 历史记录
的权重为0.1
            else:
                results.append((key_results[i][0], key_results[i][1]))
        results = sorted(results, key=lambda d: d[1], reverse=True)
    else:
        results = []
        for i in range(len_key_tfidf_url_list):
            results.append((key_results[i][0], key_results[i][1]))
        results = sorted(results, key=lambda d: d[1], reverse=True)

    return_list = []
    for result in results:
        if result[1] > 0:
```

```
            return_list.append((result[0], result[1]))
    return return_list
```

4.2 站内查询

限定查询的网站或域名(网页前缀或后缀) 比如使用网站`http://www.chinanews.com/sh`限制

4.3 短语查询

短语查询即不对关键词进行分词，完全匹配某个短语 `周末北京` 使用或不使用短语查询的对比

不使用短语查询

周末北京

找到约 275 条结果 （用时 0.48 秒） 高级搜索

周末北京晴天在线持续回暖 明天最高气温或重回冰点之上

中国天气网讯周末两天(12月23日至24日)，北京晴天在线，将持续处于回暖模式，明天最高气温可重回冰点之上，体感要比前几天暖和不少。但目前，北京部分建筑物仍有残雪，天气转暖，公众外出需注意落雪、坠冰等危险的发生。

http://www.chinanews.com/sh/2023/12-23/10133933.shtml 网页快照

【新思想引领新征程】北京城市副中心高标准建设高质量发展

央视网消息(新闻联播)：习近平总书记指出，建设北京城市副中心是北京建城立都以来具有里程碑意义的一件大事，对新时代北京的发展是一个重大机遇。规划建设7年来，北京城市副中心落实世界眼光、国际标准、中国特色、高点定位的要求，高质量发展不断提速，正成为北京这座千年古都又一张靓丽的城市名片。

http://www.chinanews.com/sh/2023/12-25/10134596.shtml 网页快照

北京城市副中心三大文化建筑周三下午开放

北京城市副中心三大文化建筑(北京艺术中心、北京城市图书馆、北京大运河博物馆)将于2023年12月27日下午对外开放。其中，北京大运河博物馆将于当日14时30分正式对公众开放。四大类型展陈、四大活动陆续开启，邀请观众登上“运河之舟”，览古今同辉。

http://www.chinanews.com/gn/2023/12-25/10134512.shtml 网页快照

“书香京城”向下扎根 全民阅读向上生长

艺起归来2023

http://www.chinanews.com/cul/2023/12-25/10134528.shtml 网页快照

北美票房：《海王2：失落的王国》力拔头筹

中新社洛杉矶12月24日电(记者张朔)《海王2：失落的王国》(AquamanandtheLostKingdom)首映力拔北美单日与周末票房头筹。

http://www.chinanews.com/cul/2023/12-25/10134608.shtml 网页快照

北京高院：民间借贷纠纷存在证据留存意识不足等风险

中新网北京12月26日电(记者陈杭)2019年以来，北京市高级人民法院(下称北京高院)共受理民间借贷纠纷再审审查案件2422件。民间借贷纠纷主要存在证据留存意识不足、虚假诉讼难以识别、法律关系多元交织、民刑交叉情况突出、合同无效情形频现、诉讼材料送达不畅等六大风险因素。

http://www.chinanews.com/sh/2023/12-26/10135703.shtml 网页快照

央视界！中央广播电视总台超高清示范园建设启动

12月23日，中央广播电视总台超高清示范园“央视界”在北京启动建设。

http://www.chinanews.com/gn/2023/12-23/10134097.shtml 网页快照

使用"周末北京"进行查询

周末北京

找到约 1 条结果 （用时 0.43 秒） 高级搜索

周末北京晴天在线持续回暖 明天最高气温或重回冰点之上

中国天气网讯周末两天(12月23日至24日)，北京晴天在线，将持续处于回暖模式，明天最高气温可重回冰点之上，体感和不少。但目前，北京部分建筑物仍有残雪，天气转暖，公众外出需注意落雪、坠冰等危险的发生。

http://www.chinanews.com/sh/2023/12-23/10133933.shtml 网页快照

4.4 通配查询

完全匹配、任意匹配、排除匹配共同组成了通配查询

任意匹配： 周末or书香

北京

找到约 10 条结果 （用时 0.41 秒） 高级搜索

“书香京城”向下扎根 全民阅读向上生长

艺起归来2023

http://www.chinanews.com/cul/2023/12-25/10134528.shtml 网页快照

周末北京晴天在线持续回暖 明天最高气温或重回冰点之上

中国天气网讯周末两天(12月23日至24日)，北京晴天在线，将持续处于回暖模式，明天最高气温可重回冰点之上，体感要比前几天和不少。但目前，北京部分建筑物仍有残雪，天气转暖，公众外出需注意落雪、坠冰等危险的发生。

http://www.chinanews.com/sh/2023/12-23/10133933.shtml 网页快照

中国A股将于除夕当日休市

中新社北京12月26日电(记者陈康亮)中国三大证券交易所26日发布2024年部分节假日休市安排，宣布将于农历除夕(2024年2月9日)日休市。

http://www.chinanews.com/cj/2023/12-26/10135666.shtml 网页快照

排除匹配： -新思想

北京

找到约 256 条结果 （用时 0.43 秒） 高级搜索

北京城市副中心三大文化建筑周三下午开放

北京城市副中心三大文化建筑(北京艺术中心、北京城市图书馆、北京大运河博物馆)将于2023年12月27日下午对外开放。其中，北京大运河博物馆将于当日14时30分正式对公众开放。四大类型展陈、四大活动陆续开启，邀请观众登上“运河之舟”，览古今同辉。

http://www.chinanews.com/gn/2023/12-25/10134512.shtml 网页快照

“书香京城”向下扎根 全民阅读向上生长

艺起归来2023

http://www.chinanews.com/cul/2023/12-25/10134528.shtml 网页快照

周末北京晴天在线持续回暖 明天最高气温或重回冰点之上

中国天气网讯周末两天(12月23日至24日)，北京晴天在线，将持续处于回暖模式，明天最高气温可重回冰点之上，体感要比前几天暖和不少。但目前，北京部分建筑物仍有残雪，天气转暖，公众外出需注意落雪、坠冰等危险的发生。

http://www.chinanews.com/sh/2023/12-23/10133933.shtml 网页快照

北京高院：民间借贷纠纷存在证据留存意识不足等风险

中新网北京12月26日电(记者陈杭)2019年以来，北京市高级人民法院(下称北京高院)共受理民间借贷纠纷再审审查案件2422件。民间借贷纠纷主要存在证据留存意识不足、虚假诉讼难以识别、法律关系多元交织、民刑交叉情况突出、合同无效情形频现、诉讼材料送达不畅等六大风险因素。

http://www.chinanews.com/sh/2023/12-26/10135703.shtml 网页快照

央视界！中央广播电视总台超高清示范园建设启动

12月23日，中央广播电视总台超高清示范园“央视界”在北京启动建设。

http://www.chinanews.com/gn/2023/12-23/10134097.shtml 网页快照

4.5 查询日志

通过cookie保存搜索历史

北京 立即搜索

找到约 257 条结果 （用时 0.43 秒） 高级搜索

【新思想引领新征程】北京城市副中心高标准建设高质量发展

央视网消息(新闻联播)：习近平总书记指出，建设北京城市副中心是北京建城立都以来具有里程碑意义的一件大事，对新时代北京的发展是一个重大机遇。规划建设7年来，北京城市副中心落实世界眼光、国际标准、中国特色、高点定位的要求，高质量发展不断提速，正成为北京这座千年古都又一张靓丽的城市名片。

http://www.chinanews.com/sh/2023/12-25/10134596.shtml 网页快照

北京城市副中心三大文化建筑周三下午开放

北京城市副中心三大文化建筑(北京艺术中心、北京城市图书馆、北京大运河博物馆)将于2023年12月27日下午对外开放。其中，北京大运河博物馆将于当日14时30分正式对公众开放。四大类型展陈、四大活动陆续开启，邀请观众登上“运河之舟”，览古今同辉。

http://www.chinanews.com/gn/2023/12-25/10134512.shtml 网页快照

“书香京城”向下扎根 全民阅读向上生长

艺起归来2023

http://www.chinanews.com/cul/2023/12-25/10134528.shtml 网页快照

周末北京晴天在线持续回暖 明天最高气温或重回冰点之上

中国天气网讯周末两天(12月23日至24日)，北京晴天在线，将持续处于回暖模式，明天最高气温可重回冰点之上，体感要比前几天暖和不少。但目前，北京部分建筑物仍有残雪，天气转暖，公众外出需注意落雪、坠冰等危险的发生。

http://www.chinanews.com/sh/2023/12-23/10133933.shtml 网页快照

相关推荐

- 北京高院：民间借贷纠纷存在证据留存意识不足等风险
- 第三届北京铸牢中华民族共同体意识研讨会举办
- 北京今起气温回升最高气温3℃ 晴天模式持续
- 北京未来三天以晴到多云为主 最高气温将升至冰点或以上
- 50辆“母亲健康快车”在北京发车 奔赴甘肃西藏等17省份
- 北京将换发第三代社保卡！能乘公交地铁能停车，功能超多

搜索历史

北京 朝鲜 北京 上海 南开 心理 铁路 周末北京

4.6 网页快照

通过爬虫时保存的html页面提供网页快照

中国新闻网 中新网 WWW.CHINANEWS.COM

即时 时政 东西问 国际 社会 财经 大湾区 华人 文体 视频 直播 图片 创意 理论

【新思想引领新征程】北京城市副中心高标准建设高质量发展

分享到:
首页 → 社会新闻
分享到:

【新思想引领新征程】北京城市副中心高标准建设高质量发展

2023年12月25日 08:18　来源：央视网
大字体
小字体
分享到:

央视网消息(新闻联播)：习近平总书记指出，建设北京城市副中心是北京建城立都以来具有里程碑意义的一件大事，对新时代北京的发展是定位的要求，高质量发展不断提速，正成为北京这座千年古都又一张靓丽的城市名片。

位于北京通州的北京城市副中心重大工程建设正加速推进，通州大运河畔，副中心“三大建筑”——北京艺术中心、北京城市图书馆、北

规划建设北京城市副中心与河北雄安新区形成北京新的“两翼”，是以习近平同志为核心的党中央作出的重大决策部署。习近平总书记高度和吸引力，科学布局生产、生活、生态空间，使工作、居住、休闲、交通、教育、医疗等有机衔接、便利快捷。

4.7 日期限制

限制时间为三天内(本报告写于12/29, 爬取网页截止至12/28)

北京

找到约 27 条结果 （用时 0.45 秒） 高级搜索

北京高院：民间借贷纠纷存在证据留存意识不足等风险

中新网北京12月26日电(记者陈杭)2019年以来，北京市高级人民法院(下称北京高院)共受理民间借贷纠纷再审审查案件2422件。民间借贷纠纷主要存在证据留存意识不足、虚假诉讼难以识别、法律关系多元交织、民刑交叉情况突出、合同无效情形频现、诉讼材料送达不畅等六大风险因素。

http://www.chinanews.com/sh/2023/12-26/10135703.shtml 网页快照

中国足协主席：坚决杜绝假球、默契球和“断子绝孙脚”

中新社北京12月26日电(记者邢翀郝凌宇)26日，中国足球协会在北京召开2024年全国青少年冬训动员会议。中国足球协会主席宋凯在会上表示，青少年足球工作要从急功近利向久久为功转变，坚决杜绝因过度追求赛事成绩导致的假球、默契球和“断子绝孙脚”等在球场上出现。

http://www.chinanews.com/ty/2023/12-26/10135723.shtml 网页快照

北京将换发第三代社保卡！能乘公交地铁能停车，功能超多

12月26日，京津冀三地在石家庄签署《京津冀社会保障卡居民服务“一卡通”合作框架协议》，共同推进京津冀“一卡通”建设，加快实现三地社保卡跨省通用、一卡多用、线上线下场景融合发展。北京市民在换发第三代社保卡后，将可享受京津冀三地交通、文旅等多个场景的服务。

http://www.chinanews.com/sh/2023/12-26/10135660.shtml 网页快照

50辆“母亲健康快车”在北京发车 奔赴甘肃西藏等17省份

中新网北京12月26日电(记者余湛奕)中国妇女发展基金会与中国建设银行26日在北京举行2023年度建行“母亲健康快车”捐赠暨发车仪式。由中国建设银行向中国妇女发展基金会捐赠的50辆“母亲健康快车”启程奔赴甘肃、西藏、新疆等17个省份，为当地妇女儿童提供健康服务，助力夯实妇幼健康基层服务网络。

http://www.chinanews.com/sh/2023/12-26/10135708.shtml 网页快照

4.8 标题搜索

限制只在标题内搜索

北京

找到约 16 条结果 （用时 0.08 秒） 高级搜索

台青北京种莓记

中新社北京12月24日电题：台青北京种莓记

http://www.chinanews.com/gn/2023/12-24/10134359.shtml 网页快照

北京城市副中心三大文化建筑周三下午开放

北京城市副中心三大文化建筑(北京艺术中心、北京城市图书馆、北京大运河博物馆)将于2023年12月27日下午对外开放。其中，北京大运河博物馆将于当日14时30分正式对公众开放。四大类型展陈、四大活动陆续开启，邀请观众登上“运河之舟”，览古今同辉。

http://www.chinanews.com/gn/2023/12-25/10134512.shtml 网页快照

北京人艺年度历史大戏《张居正》登台

中新网北京12月23日电(记者高凯)北京人艺日前迎来跨年大戏——原创历史剧《张居正》的首演。

http://www.chinanews.com/cul/2023/12-23/10134103.shtml 网页快照

北京高院：民间借贷纠纷存在证据留存意识不足等风险

中新网北京12月26日电(记者陈杭)2019年以来，北京市高级人民法院(下称北京高院)共受理民间借贷纠纷再审审查案件2422件。民间借贷纠纷主要存在证据留存意识不足、虚假诉讼难以识别、法律关系多元交织、民刑交叉情况突出、合同无效情形频现、诉讼材料送达不畅等六大风险因素。

http://www.chinanews.com/sh/2023/12-26/10135703.shtml 网页快照

北京凌锋公益基金会向柬埔寨西哈努克省捐建卫生站

中新网金边12月25日电(杨强黎海月)由北京凌锋公益基金会捐款修建的柬埔寨长龄卫生站，23日在西哈努克省甘榜塞拉县举行落成典礼。西哈努克省官员以及500余名当地居民出席剪彩启用仪式。

http://www.chinanews.com/gj/2023/12-25/10134962.shtml 网页快照

5. 个性化查询

根据用户的历史搜索记录提供个性化查询。将近期历史搜索记录也进行分词并计算与文档的余弦相似度，其占最终余弦相似度得分的0.1

```
if len(history_words) > 0:  # 没有历史记录时不计算历史记录的相似度
    history_results_dict = {}
    for i in key_results_index:  # 遍历每个文档
        num = 0
        _key_tfidf_list = key_tfidf_list[i]
        for _key_history in key_history:  # 遍历每个关键输入词
            if _key_history[1] != 0:
                for __key_tfidf_list in _key_tfidf_list:  # 遍历每个文档内的每个关键词
                    if _key_history[0] == __key_tfidf_list[0]:  # 若为相同单词
                        num = num + _key_history[1] * __key_tfidf_list[1]
        history_results_dict[i] = ((key_tfidf_url_list[i], (round(num / (len_key_history *
length(_key_tfidf_list)), 4))))  # 存储第i个文档的余弦相似度

    results = []
    for i in range(len_key_tfidf_url_list):
        if key_results[i][1] == 0:
            pass
```

```
        elif j := history_results_dict.get(i):
            results.append((key_results[i][0], key_results[i][1] + j[1] / 10))  # 历史记录的权
重为0.1
        else:
            results.append((key_results[i][0], key_results[i][1]))
    results = sorted(results, key=lambda d: d[1], reverse=True)
else:
    results = []
    for i in range(len_key_tfidf_url_list):
        results.append((key_results[i][0], key_results[i][1]))
    results = sorted(results, key=lambda d: d[1], reverse=True)
```

6. web页面

使用flask框架构建web页面

7. 个性化推荐

由于中国新闻网的url中已经包含了新闻的分类，所以我们无需对文档进行聚类等操作。

1. 用户浏览某个网页，根据该新闻类别更新用户画像(对应类别value+1)
2. 每次搜索时，根据搜索的关键词和用户画像推荐相关新闻

推荐规则：

1. 假如要推荐num个新闻，用户画像中每种类别m的value为m_value，则(m_value/总value)*num为推荐m类别新闻的数目
2. 根据搜索的关键词返回的结果里抽取对应类别的新闻
3. 如果结果不足则从对应类别中抽取相关性没有那么高的新闻

```
def _suggest():

    keywords = request.args.get('keywords')
    if not keywords:
        ret_list = []
        return jsonify(ret_list)

    up = get_user_profile('./user_profile.csv')
    search_history = []

    try:
        result_list: list[tuple[str, float]] = main(keywords, search_history,True)
        print(result_list)
    except KeyError :
        try:
            result_list: list[tuple[str, float]] = main(keywords, search_history,False)
            print(result_list)
        except KeyError:
            result_list = []

    sug_url_list = []
    if not result_list:
        nums = 6
        for category, percentage in up.items():
            temp_list = get_url_list(category, percentage*nums)
            sug_url_list.extend(temp_list)
    else:
        temp_list = []
        res_list = calculate_ratio(result_list)
        for item in res_list:
            url = item[0]
            sc_1 = item[1]
            cate = (url_title_df.loc[url])['category']
            title = (url_title_df.loc[url])['title']
            if cate in up:
                sc_2 = up[cate]
            else:
                sc_2 = 0
            sc = 2*sc_1 + sc_2
            temp_list.append({
                'title':title,
                'url':url,
                'score':sc
            })
        sorted_temp_list = sorted(temp_list, key=lambda x: x['score'], reverse=True)
        sug_url_list = [{k: v for k, v in d.items() if k != 'score'} for d in
sorted_temp_list[:6]]

    ret_list = []
    for item in sug_url_list:
        _url = item['url']
        _title = item['title']
        sug = f'<a href="{_url}">{_title}</a>'
        ret_list.append(sug)

    return jsonify(ret_list)
```