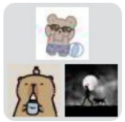
自然语言处理研讨课
（实践课）
第3章 文本爬取和处理实践

赵 阳
中国科学院自动化研究所
yang.zhao@nlpr.ia.ac.cn

课程网站

课程网站：



群聊：自然语言处理研讨课-2025

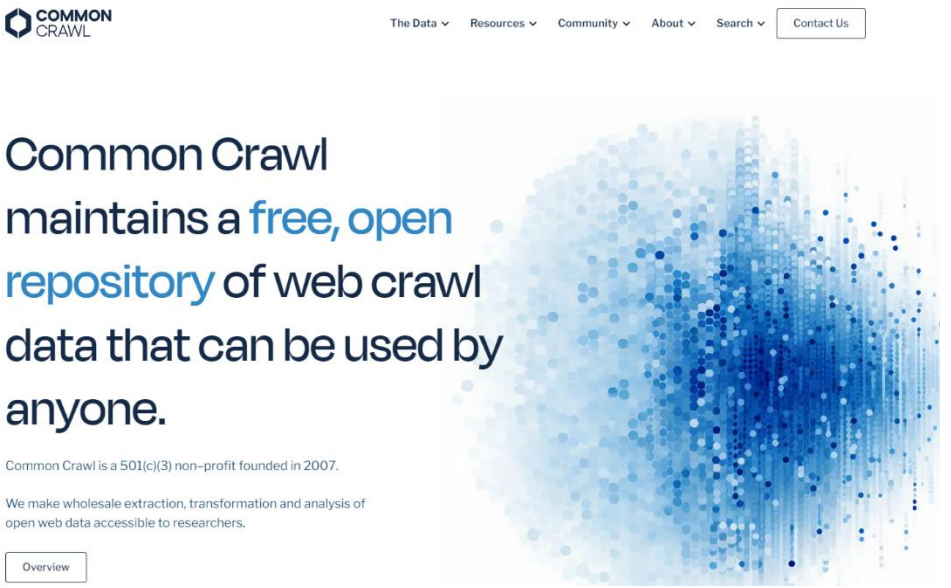
该二维码7天内(10月6日前)有效，重新进入将更新

本章内容

1. 网络爬虫和数据爬取
2. 文本数据处理
3. 本章实践

1. 网络爬虫和数据爬取

- ChatGPT/GPT-3模型的文本数据：
  CommonCrawl数据集
- 互联网中的网络数据对NLP和大模型而言有着至关重要的意义。
- 它提供了丰富的、多样化的海量数据资源，包括新闻报道、社交媒体、学术论文等，涵盖了广泛的语言现象。



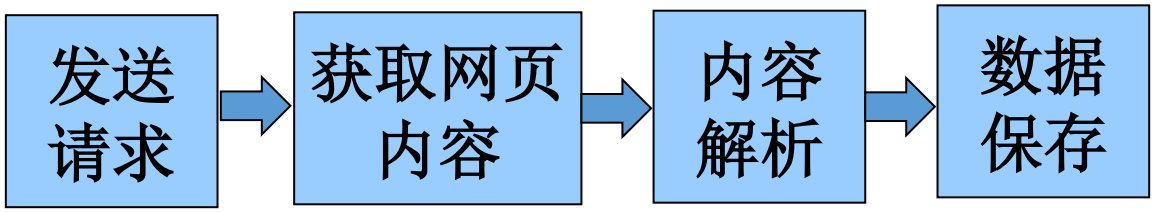
掌握网络数据的爬取和处理对于从事自然语言处理的学术研究或产品开发等都是一项基本的技能。

1. 网络爬虫和数据爬取

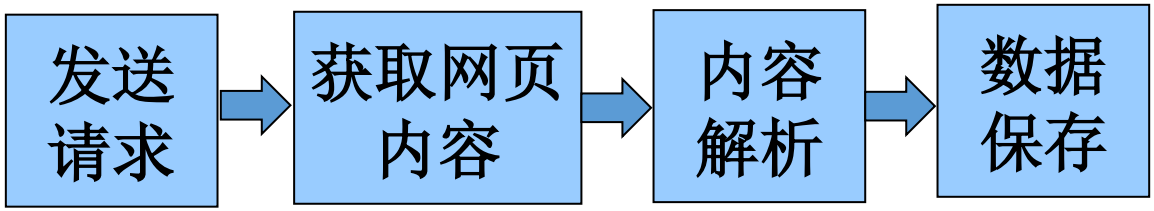
■ 网络爬虫

网络爬虫是一种按照一定的规则，自动地抓取网络中信息的程序

■ 基本过程



1. 网络爬虫和数据爬取



1、发送请求

使用http库向目标站点发起请求。

2、获取网页内容

如果请求内容存在于目标服务器上，那么服务器会返回请求内容。

返回内容包含：html、图片和视频等。

3、内容解析

解析html数据，利用正则表达式或者其他库提取目标信息。

第三方解析库如Beautifulsoup等

4、数据保存

解析得到的数据保存在本地。

1. 网络爬虫和数据爬取

www.news.cn 新华网 NEWS www.xinhuanet.com
新华通讯社主办 公司官网 股票代码：603888

学习进行时 高层 时政 人事 国际 财经 网评 港澳 台湾 思客智库 全球连线 教育 科技 科普 体育 文化 健康 军事 访谈 视频 图片 中央文件
金融 汽车 食品 房产 信息化 乡村振兴 溯源中国 城市 旅游 能源 会展 彩票 娱乐 时尚 悦读 公益 书画 一带一路 亚太网 上市公司 投教基地

新华网 > 科技 > 正文

2023
10/29
22:55:52
来源：新华网

东风着陆场做好各项准备迎接神舟十六号回家

字体： 小 中 大 分享到：

新华社酒泉10月29日电（李秉宣、张艳）神舟十六号航天员乘组将于10月31日返回东风着陆场。记者29日晚从东风着陆场了解到，目前，着陆场已做好各项准备工作，等待神十六乘组天外归来。

这是东风着陆场执行的第5次载人飞船搜索和航天员救援任务。

10月29日下午，神舟十六号、神舟十七号两个航天员乘组在中国空间站里进行了交接。之后，神舟十六号航天员乘组将根据计划返回东风着陆场。

据介绍，针对这次任务特点，东风着陆场开展了大量针对性准备工作——组建3支专业搜救力量和4支支援保障分队，协同完成搜救任务。在后弹道返回着陆区、推迟一圈返回着陆区，地面搜救小组、着陆场区周边数十个民兵分队为专业搜救力量提供支援。按照单项训练、系统间匹配训练、空地协同训练、全系统演练等4个阶段组织了训练演练。

【责任编辑:张欣然】

新华网客户端

分享到

1. 网络爬虫和数据爬取

- 核心的程序

```
import requests

def getHTMLText(url):  #python函数
    try:
        r = requests.get(url)  #发送请求
        r.encoding = r.apparent_encoding  #获取相应内容编码
        return r.text
    except:
        print('爬取失败')


if __name__ =='__main__':
    url = 'https://www.news.cn/tech/2023-10/29/c_1129946917.htm'
    print(getHTMLText(url))
```

核心程序

1. 网络爬虫和数据爬取

```
(base) PS D:\python-NLP> python .\test_requests.py
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /><meta name="publishid" content="1129946917.12.103.0"
983.1129946917"/>
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-mobile-web-app-status-bar-style" content="black" />
<meta content="telephone=no" name="format-detection" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta content="width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0,user-scalable=no" name="viewpo
<meta name="keywords" content="东风着陆场,迎接神舟十六号回家
" />
<meta name="description" content="东风着陆场做好各项准备迎接神舟十六号回家
---神舟十六号航天员乘组将于10月31日返回东风着陆场。记者29日晚从东风着陆场了解到，目前，着陆场已做好各项准备工作，等待神十
<fjtignoreurl>
<script src="http://www.news.cn/global/js/pageCore.js"></script>
<title>
```

运行结果（部分）

1. 网络爬虫和数据爬取

r.text的内容较多，但真正需要的内容仅占很小一部分，因此有两种方式提取需要的内容：

1） 利用规则

（<p>和 <\p>之间的内容）

<p>和<\p>为段落标记

```
<div id="detail">
<p>    新华社酒泉10月29日电（李秉宣、张艳）神舟十六号航天员乘组将于10月31日返回东风着陆场
<p>    这是东风着陆场执行的第5次载人飞船搜索和航天员救援任务。</p>
<p>    10月29日下午，神舟十六号、神舟十七号两个航天员乘组在中国空间站里进行了交接。之后，
<p>    据介绍，针对这次任务特点，东风着陆场开展了大量针对性准备工作—组建3支专业搜救力量
。按照单项训练、系统间匹配训练、空地协同训练、全系统演练等4个阶段组织了训练演练。</p>
<div id="articleEdit">
```

1. 网络爬虫和数据爬取

2） 利用现有的库(BeautifulSoup)

```
import requests
from bs4 import BeautifulSoup
def getHTMLText(url):  ##python函数
        try:
          news=''
          r = requests.get(url)  ##发送请求
           r.encoding = r.apparent_encoding  ##获取相应内容编码
           soup = BeautifulSoup(r.text, 'html.parser')
          title = soup.title.text.strip()  ##获得标题
           title +='!!!!'  ##区分标记
           news += title
          for x in soup.find_all('div', {'id': ['detail']}):
              for y in x.find_all('p'):
                text = y.text.strip()
                news += text
         return news
       except:
          print('爬取失败')
```

1. 网络爬虫和数据爬取

■ 获取大量数据

1) 查找url的规律

像百度百科、京东商品和豆瓣影评等URL是有规律的

```
for i in range(max_num) :
    url = "https://baike.baidu.com/view/" + str(i) + ".htm"
    try:
        text = getHTMLText(url)
    except:
        print(“爬取失败”)
```

1. 网络爬虫和数据爬取

2）以某个网页(例如新华网页面) 为种子，找到该网站所有的超链接（href），再去爬取每个网页

超链接

```
import requests
from bs4 import BeautifulSoup
def getHTML (url):  ##python函数
    try:
        news_list = []#空列表
        r = requests.get(url)  ##发送请求
        r.encoding = r.apparent_encoding  ##获取相应内容编码
        soup = BeautifulSoup(r.text, 'html.parser')
        tags = soup.find_all('a')  #找到所有锚/超链接
        for tag in tags:
            news_list.append((str(tag.get('href')).strip()))  ##得到href
        return news_list
    except:
        print('爬取失败')

if __name__ =='__main__':
    url = 'https://www.news.cn/'
    print(getHTML(url))
```

1. 网络爬虫和数据爬取

主要的逻辑为：

- 写文件
- url列表

- 读文件内容

```
if __name__ == '__main__':
    home_url = "http://www.xinhuanet.com/worldpro/"
    fo = open('xinhua_news.txt', "w",encoding="utf-8")
    url_list = getHTML(home_url) # 得到所有url列表
    url_list = list(set(url_list))  #去重
    for url in url_list:
        news = getText(url)  # 得到每个url的内容
        if news == None:  # 去掉为空的内容
            continue
        fo.write(news+'\n')
        sub_url_list = getHTML(url)  # 得到所有url列表
        if sub_url_list !=None:
            sub_url_list = list(set(sub_url_list))  # 去重
            for sub_url in sub_url_list:
                news = getText(sub_url)  # 得到每个url的内容
                if news == None:  # 去掉为空的内容
                    continue
                fo.write(news + '\n')
    fo.close()
```

本章内容

1. 网络爬虫和数据爬取
2. 文本数据处理
3. 本章实践

2. 文本数据处理

■ 分词（word segmentation/ tokenization/）：

- 将文本切分成一个**单词**序列
- 西方屈折语（英语、法语和德语等），词与词之间有空格之类的显式标志指示词的边界；
- 孤立语和黏着语 (如汉语、日语和越南语等)，词与词之间没有空格。

2. 文本数据处理

- 中文分词：

采用的工具jieba

详细介绍见：https://github.com/fxsjy/jieba

安装 conda install jieba

```
import jieba

f_in=open('xinhua_news.txt','r',encoding="utf-8")
f_out=open('xinhua_news.txt.seg','w',encoding="utf-8")
lines = f_in.readlines()
for i in lines:
    seg_list=list(jieba.cut(i, cut_all=False))
    if len(seg_list)>=10:
        f_out.write(' '.join(seg_list))
f_in.close()
f_out.close()
```

2. 文本数据处理

■ 正则表达式：

- 字符串处理时，往往要根据特定规则处理对应模式和字符串（如日期、邮箱、电话等）。
- 正则表达式能够方便地利用规则或者模板查找特定模板。
- Python官网的正则表达式说明：https://docs.python.org/zh-cn/3/library/re.html

（大家可以自学）

2. 文本数据处理

- 正则表达式举例（找到字符串中的日期）：

```
import re

f_in=open('xinhua_news.txt','r',encoding="utf-8")
lines = f_in.readlines()
pattern = r'\d{4}-\d{2}-\d{2}|\d{4}年\d{1,2}月\d{1,2}日|\d{1,2}月\d{1,2}日'
for i in lines:
    results=re.findall(pattern, i)
    if len(results)>1:
        print(results)
f_in.close()
```

```
\d{4}-\d{2}-\d{2}：
\d{4}：匹配 4 位数字。
匹配 "年 - 月 - 日" 格式的日期
\d{4}年\d{1,2}月\d{1,2}日
匹配 "年月日" 中文格式的日期
\d{1,2}月\d{1,2}日
匹配 "月日" 中文格式的日期
```

2. 文本数据处理

```
(python-NLP) PS D:\python-NLP> python .\test_pattern.py
['10月29日', '2023年10月15日', '10月23日', '10月23日']
['10月28日', '10月28日', '10月28日', '10月28日']
['10月25日', '10月25日']
['10月28日', '10月28日', '10月28日']
['10月20日', '10月21日', '10月22日']
['10月29日', '10月28日']
['10月25日', '10月26日', '10月25日', '10月25日', '10月25日']
['10月27日', '7月21日', '8月15日', '9月15日']
['10月29日', '10月7日', '10月28日', '10月28日', '10月16日', '10月11日', '10月27日']
['10月27日', '2013年4月27日']
['10月25日', '10月25日']
['10月29日', '10月29日']
```

结果

本章内容

1. 网络爬虫和数据爬取
2. 文本数据处理
3. 本章实践

本节实践

■ 数据爬取和处理实践基本要求：

- 利用爬虫从网络分别爬取中文和英文数据

  中英文不少于5M，不多于10M，能够上传到课程网站即可

- 爬取结束后进行基本的处理

  例如中文分词、英文tokenize

- 抽取文本数据中的数字和日期
- 统计并画出中文和英语单词的词频分布，验证齐夫定律

注意：本次作业需要提交，本节课不用提交，**与下节课的文本表示一起提交**，提交时间为10月7号之前（待定）

提交内容：实践报告、代码和数据

本节实践

■ 要求和评分准则：

- 报告需要详细说明爬取和处理的基本方法、数据来源、算法流程、分析结果、结论等；
- 不需要把程序大段复制上去，如果有需要，只需要截取和复制关键程序即可；
- 报告页数不需要太长。
- 如果满足了基本要求就可以得到B（80-85分）
- 剩余的15分， 看大家的自己拓展分析和实验

不同网站和主题的词频差异等等

不同方法得到词向量的差别

不同主题下词向量的差别

…

谢谢！

Thanks!

下次上课时间：10月14日（周二）