实验目的：从网络上获取新闻与评论等所需数据，做好项目的数据准备工作

实验工具：Python，格式转换工具（在线小工具：JSON转Excel）

实验过程：

第一部分的实验目的是为项目提供数据样本。通过讨论，我们组决定从央视新闻的微博账户中获取本次实验所需的数据。我们通过爬虫技术从央视新闻的微博账户中获取了2019年12月至2020年6月的新闻。对于每一条新闻，我们关心并截取的数据如下：新闻本身的内容（标题也包含在其中）、评论数、以及评论的具体内容。获取完所有数据之后，通过格式转换工具将JSON格式转换为Excel格式。表格的每一行记录了一条新闻的所有数据，其中，第一列记录的是新闻本身的文本内容，第二列记录的是评论总数，第三列及以后每一列记录一条评论。最后，按大作业要求的时间段划分将数据分组并整理成4张Excel表格，并调整格式，项目数据准备阶段结束。

实验分析：

选择央视新闻的微博账户作为数据来源的理由是：第一，央视新闻是我国最重要的官方新闻舆论机构，具有极高的公信力，新闻的准确性能得到最大限度的保证。第二，央视新闻拥有规模庞大的受众人群，所以评论的数量和质量非常满足实验需求。

通过对网址规律的分析，我们选择按天爬取新闻内容，并在前十页评论中随机抽取六页进行保存。在爬虫的过程中，我们遭遇到了各种未知的报错和异常，在处理完异常之后会出现因数据缺失而形成的空白。为了消除这些空白对后续工作可能会造成的影响（比如有些文本分析的算法会把空白作为结束的判断依据），我们将会对Excel表格中的空白进行特殊处理。如果某一栏评论数数据为空（也就是该新闻的全部评论都获取失败了），我们会在这一栏填补数字0，如果某一列评论内容为空（也就是仅这一条评论获取失败），我们会在这一栏填补特殊语段“我是只大大龙”加以区分。

实验代码：

# -\*- coding: utf-8 -\*-  
import random  
import requests  
from bs4 import BeautifulSoup  
import re  
import json  
import os  
import time  
from lxml import html  
etree = html.etree  
  
  
class Weibospider:  
 def \_\_init\_\_(self, date):  
 # 获取首页的相关信息：  
 # self.start\_url = 'https://weibo.com/2803301701/Is5xeqvl1?from=page\_1002062803301701\_profile&wvr=6&mod=weibotime&type=comment#\_rnd1608714031543'  
 # self.start\_url = 'https://weibo.com/rmrb?is\_all=1&stat\_date=201912&page={0}#1608738638854'  
 self.start\_url = 'https://weibo.com/p/aj/v6/mblog/mbloglist?ajwvr=6&domain=100206&is\_all=1&stat\_date=2020' + date + '&page={0}&pagebar=0&pl\_name=Pl\_Official\_MyProfileFeed\_\_27&id=1002062803301701&script\_uri=/rmrb&feed\_type=0&pre\_page=0'  
 self.headers = {  
 "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8",  
 "accept-encoding": "gzip, deflate, br",  
 "accept-language": "zh-CN,zh;q=0.9,en;q=0.8",  
 "cache-control": "max-age=0",  
 "cookie": "\_s\_tentry=link.csdn.net; Apache=5388654551837.863.1611116180656; SINAGLOBAL=5388654551837.863.1611116180656; ULV=1611116180667:1:1:1:5388654551837.863.1611116180656:; login\_sid\_t=aa54a96816b4293c377add075a9d0bf8; cross\_origin\_proto=SSL; WBStorage=8daec78e6a891122|undefined; wb\_view\_log=1280\*7201.5; crossidccode=CODE-yf-1L26ra-26rvfy-fKA3ME0np2r41Pzf88e1e; SSOLoginState=1611122034; SUB=\_2A25NA7UiDeRhGeBK7FYT8SvJyT6IHXVuD9tqrDV8PUJbkNAKLUfSkW1NR4RUrDbcgsWkIaSKjBAegNEPZYyuKP5t; SUBP=0033WrSXqPxfM725Ws9jqgMF55529P9D9W5BoHBrsMgdLmOuMzyL8ZPd5NHD95QcShMXeo2fSKzEWs4Dqcj.i--RiKyFiKysi--fi-82iK.7i--4iK.Ri-isi--NiK.0i-2E; wvr=6; UOR=link.csdn.net,open.weibo.com,graph.qq.com; wb\_view\_log\_6474215522=1280\*7201.5; webim\_unReadCount=%7B%22time%22%3A1611122074058%2C%22dm\_pub\_total%22%3A0%2C%22chat\_group\_client%22%3A0%2C%22chat\_group\_notice%22%3A0%2C%22allcountNum%22%3A38%2C%22msgbox%22%3A0%7D",  
 "referer": "https://www.weibo.com/u/5644764907?topnav=1&wvr=6&topsug=1",  
 "upgrade-insecure-requests": "1",  
 "user-agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.96 Safari/537.36",  
 }  
 self.date = date  
  
 def parse\_home\_url(self, url): # 处理解析首页面的详细信息  
 start = time.time()  
 #print(url)  
 res = requests.get(url, headers=self.headers)  
 while time.time() - start > 5:  
 print("响应超时，重新获取")  
 start = time.time()  
 res = requests.get(url, headers=self.headers)  
 response = res.content.decode().replace("\\", "")  
 every\_id = re.compile('name=(\d+)', re.S).findall(response) # 获取次级页面需要的id  
 home\_url = []  
 for id in every\_id:  
 base\_url = 'https://weibo.com/aj/v6/comment/big?ajwvr=6&id={}&from=singleWeiBo'  
 url = base\_url.format(id)  
 home\_url.append(url)  
  
 res.encoding = res.apparent\_encoding  
 soup = BeautifulSoup(json.loads(res.text)["data"], "html.parser")  
 titles = soup.find\_all("div", class\_="WB\_text W\_f14") # 获取每条微博的内容  
 title\_list = []  
 for title in titles:  
 title\_list.append(title.text.replace("\n", "").replace(" ", ""))  
  
 times = re.compile('fromprofile"> (.\*?)</a>').findall(response) # 获取每条微博的时间  
 dates = []  
 for time\_ in times:  
 month = time\_.split("-")[1].zfill(2)  
 day = time\_.split("-")[2].split(" ")[0].zfill(2)  
 dates.append("2020-" + month + "-" + day)  
 return dates, title\_list, home\_url  
  
 def parse\_comment\_info(self, url): # 爬取直接发表评论的人的相关信息(这里只关注评论信息)  
 res = requests.get(url, headers=self.headers)  
 response = res.json()  
 count = response['data']['count']  
 counting = 0  
 while count is None:  
 counting+=1  
 print("[error] 获取评论出错!!!重新获取。")  
 res = requests.get(url, headers=self.headers)  
 response = res.json()  
 count = response['data']['count']  
 if counting>50:  
 break  
 if count == 0:  
 return None  
 html = etree.HTML(response['data']['html'])  
 name = html.xpath("//div[@class='list\_li S\_line1 clearfix']/div[@class='WB\_face W\_fl']/a/img/@alt") # 评论人的姓名  
 info = html.xpath("//div[@node-type='replywrap']/div[@class='WB\_text']/text()") # 评论信息  
 info = "".join(info).replace(" ", "").split("\n")  
 info.pop(0)  
 comment\_time = html.xpath("//div[@class='WB\_from S\_txt2']/text()") # 评论时间  
 name\_url = html.xpath("//div[@class='WB\_face W\_fl']/a/@href") # 评论人的url  
 name\_url = ["https:" + i for i in name\_url]  
 comment\_info\_list = []  
 for i in range(len(name)):  
 item = {}  
 item["comment\_info"] = info[i] # 存储评论的信息  
 comment\_info\_list.append(info[i].replace("：", ""))  
 return count, comment\_info\_list  
  
 def write\_file(self, path\_name, data):  
 if os.path.exists(path\_name):  
 with open(path\_name, "r", encoding='utf-8') as f:  
 data\_0 = json.load(f)  
 data\_0.append(data)  
 with open(path\_name, "w+", encoding='utf-8') as f:  
 json.dump(data\_0, f, ensure\_ascii=False, indent=4)  
 else:  
 with open(path\_name, "w+", encoding='utf-8') as f:  
 data\_0 = [data]  
 json.dump(data\_0, f, ensure\_ascii=False, indent=4)  
  
 def run(self):  
 start\_url = self.start\_url  
 start\_ajax\_url1 = 'https://weibo.com/p/aj/v6/mblog/mbloglist?ajwvr=6&domain=100206&is\_all=1&stat\_date=2020{1}&page={0}&pagebar=0&pl\_name=Pl\_Official\_MyProfileFeed\_\_27&id=1002062803301701&script\_uri=/rmrb&feed\_type=0&pre\_page={0}'  
 start\_ajax\_url2 = 'https://weibo.com/p/aj/v6/mblog/mbloglist?ajwvr=6&domain=100206&is\_all=1&stat\_date=2020{1}&page={0}&pagebar=1&pl\_name=Pl\_Official\_MyProfileFeed\_\_27&id=1002062803301701&script\_uri=/rmrb&feed\_type=0&pre\_page={0}'  
 i = 0  
 while True:  
 i += 1  
 if (i>=500):  
 break  
 print("[info] 获取第 " + str(i) + " 页微博。")  
 dates\_1, titles\_1, home\_url = self.parse\_home\_url(start\_url.format(i, self.date)) # 获取每一页的微博  
 if len(home\_url) == 0:  
 print("[waring] 最后一页是：第 " + str(i-1) + " 页")  
 break  
 dates\_2, titles\_2, ajax\_url1 = self.parse\_home\_url(start\_ajax\_url1.format(i, self.date)) # ajax加载页面的微博  
 dates\_3, titles\_3, ajax\_url2 = self.parse\_home\_url(start\_ajax\_url2.format(i, self.date)) # ajax第二页加载页面的微博  
 all\_url = home\_url + ajax\_url1 + ajax\_url2  
 all\_dates = dates\_1 + dates\_2 + dates\_3  
 # print(all\_dates)  
 all\_titles = titles\_1 + titles\_2 + titles\_3  
  
 for j in range(len(all\_url)):  
 try:  
 print("[info] 获取第 " + str(i) + " 页第 " + str((i-1) \* 45 + j + 1) + " 条微博的评论。")  
 path\_name = "temp01/{0}.json".format(all\_dates[j])  
 print("[info] 获取第 " + str(i) + " 页第" + str(j + 1) + "条，总第 " + str((i - 1) \* 45 + j + 1) + " 条微博第 1 页的评论。")  
 all\_count, comment\_info\_list = self.parse\_comment\_info(all\_url[j])  
 # self.write\_file(path\_name, comment\_info\_list)  
 data = {"微博内容": all\_titles[j], "评论数量": all\_count, "评论": comment\_info\_list}  
 except Exception as e:  
 print("Error", e)  
 time.sleep(10)  
  
 LisT = range(1,10)  
 py = random.sample(LisT,5)  
 for num in py:  
 try:  
 print("[info] 获取第 " + str(i) + " 页第" + str(j + 1) + "条，一共第 " + str((i - 1) \* 45 + j + 1) + " 条微博第 " + str(num + 1) + " 页的评论。")  
 except Exception as e:  
 print("Error:", e)  
 time.sleep(10)  
 try:  
 if num \* 15 < int(all\_count) + 15:  
 comment\_url = all\_url[j] + "&page={}".format(num + 1)  
 # print(comment\_url)  
 try:  
 count, comment\_info\_list = self.parse\_comment\_info(comment\_url)  
 # self.write\_file(path\_name, comment\_info\_list)  
 # data["评论数量"] = data["评论数量"] + count  
 data["评论"] = data["评论"] + comment\_info\_list  
 except Exception as e:  
 print("Error:", e)  
 time.sleep(10)  
 count, comment\_info\_list = self.parse\_comment\_info(comment\_url)  
 # self.write\_file(path\_name, comment\_info\_list)  
 # data["评论数量"] = data["评论数量"] + count  
 data["评论"] = data["评论"] + comment\_info\_list  
 del count  
 time.sleep(0.2)  
 except Exception as e:  
 print("Error:", e)  
 time.sleep(10)  
 self.write\_file(path\_name, data)  
 print(data)  
 print("[info] 第{}条微博信息获取完成！写入文件:".format((i-1) \* 45 + j + 1) + path\_name)  
 # print("第{}条微博信息获取完成！".format((i-1) \* 45 + j + 1))  
 # break  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 weibo = Weibospider("06")  
 weibo.run()

\*注：该代码实现的是对2020年6月数据的爬取。如果要爬取2020年其它月份的数据，只要在main中把Weibospider的参数做出对应修改即可。如果要爬取2019年的数据，仅需要把过程函数run中的变量start\_ajax\_url1和start\_ajax\_url2修改即可。

结果呈现：在“爬虫部分的代码以及全部数据”文件夹中保存有全部的过程数据和最终的结果数据。