****

课程设计报告书

**题目: 5G手机评价分析**

**学 院 计算机科学与工程学院**

**专 业 计算机科学与技术（全英创新班）**

**学生姓名 彭鹏、林师言、陈秉亮、肖土圣**

**学生学号201830570231、201830570194、201830570019、201830570347**

**指导教师 李剑**

|  |  |
| --- | --- |
| 教  师  评  语 | 教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |

5G手机评价分析

一、选题背景

进入5G时代，各大手机厂商纷纷推出5G手机，为了真实评估各主流的5G手机的用户口碑，可以登陆到主流电商网站，根据用户评价，识别出用户反馈的优劣势以及关注倾向度，从而客观评价5G手机的真实反馈。

登陆主流电商网站，锁定主流的5G手机，分别获取各款手机完整的商品评价，通过对文本解析和分词处理，并剔除异常数据，汇总后得出各款手机的优点和缺点，为消费者提供选型参考。

本课题利用消息队列、Redis及Elastic Search实现了爬虫任务的生成、分发及查询，通过代理池向京东商品评论接口进行请求，并通过使用NLP对商品评论的好坏进行分类、使用jieba实现分词和词频进行分析，实现了从商品名到商品评价分析的全流程服务端。

二、方案论证(设计理念)

1. 爬虫系统设计

网络爬虫是指通过对特定网页或接口发起请求，获取相应页面或返回值，从中筛选出需要的信息并且进行持久化存储，选择性交由后续进行数据分析处理。在该设计中，根据Chrome DevTool 分析得出，其评论的获取可通过图一的API接口获取，其中通过GET参数请求传入商品ID、评论类型、排序方式和分页即可获得对应评论信息。

图形用户界面, 文本

描述已自动生成

图一： 京东评论接口及参数

根据测试发现，该接口对IP访问频率限制了访问次数，要稳定地进行访问，请求间隔需要设置在2s以上，严重限制评论信息爬取效率。同时，根据网站robot.txt描述，带有特定User-Agent的请求会被阻拦。因此，本设计爬虫部分采取了以下几种措施。

第一，使用IP代理池配置。通过更换代理IP进行请求以避开接口对IP访问频率的限制。

第二，配置特定的request header以模拟浏览器访问请求。

第三，采取任务分发的模式，利用消息队列，将爬虫任务由服务端分发到各爬虫端以进行分布式爬取。

通过以上设计，爬虫避免了每次请求之间的2s延迟，爬取效率能得到显著提高。

1. 服务端系统设计

由于采用了任务分发机制，该设计通过服务端控制进行任务的生成、分发、查询及删除。 服务端采用GIN进行搭建，向外提供任务的增删改查接口。

* 1. 任务生成

根据用户需求不同，该接口支持根据商品ID或商品名进行搜索。对于以商品ID为关键词的情况，需爬取商品详情页获取其对应商品名；对于以商品名作为关键词的情况，需爬取商品搜索页进行搜索，选择数件商品，并对每件商品生成对应任务。同时，利用redis进行商品ID维度去重。生成后交付消息队列分发至爬虫端。

* 1. 任务分发

任务分发部分采用redis模拟消息队列，支持多端输入及多端输出，理论上可供给多个服务端对大量请求进行分流及多个爬虫端获取任务进行处理。

* 1. 任务搜索

任务搜索部分采用ElasticSearch搜索引擎，使用ik\_max\_word分词器支持中文商品名分词。通过该组件实现了任务的模糊搜索。

* 1. 任务删除

任务删除需进行延期删除，以确保任务能从MySQL、ElasticSearch、Cache及消息队列中完全删除，避免出现任务Id无法找到对应任务的情况。

* 1. 异常数据定时监测任务

由于爬虫端及数据分析端可能出现异常情况导致任务处理不完全，需定时对任务列表进行搜索排查，将异常任务相关数据清除并重新生成，重新提交至爬虫侧。

* 1. 定时提交任务

对于用户生成的多个任务，服务端以一定时间长度为间隔，定时扫描任务列表，将未提交的任务提交至消息队列进行处理。同时提供手动接口至用户，支持主动提交任务。

1. 数据分析系统设计

数据分析系统能够完成从数据清洗，到分词，到词频统计和词云生成这一系列的工作。

3.1 数据清洗

数据清洗主要做的是数据去重，避免重复的无意义的商品评论对统计结果的影响。

3.2 自然语言处理

使用SnowNLP把商品评论分为好评和差评，以便后续步骤统计商品的优缺点。

3.3 分词

使用jieba库对商品评价做分词并筛选形容词作为有效的商品评价词汇，以便后续进行词频统计。

3.4 词频统计

使用Counter函数统计词频然后转成json形式的数据存入MySQL中。

3.5 生成词云

使用WordCloud根据分词后的商品评论分别生成好评和差评的词云，直观展现商品的优缺点。

3.6 统计用户关注倾向（陈秉亮写）

（正文格式：宋体，小4号，不加粗，两端对齐，行距为固定值20磅）

三、过程论述

1. 服务端设计及实现
   1. 任务的实现和流转
      1. 任务的定义

任务是指对于某一种需要爬取的商品对应生成的一个实体，其中包含对该商品进行爬虫处理的信息以及数据分析完成后的结果。对于用户的精确搜索，即给定商品ID的搜索，根据此ID对应生成一个任务。对于用户的模糊搜索，即给定关键词（手机型号）的搜索，会根据关键词查询得一系列商品并生成多个任务。

* + 1. 任务的存在形式

在系统中，任务需在MySQL，Redis（Message Queue）和Elastic Search之间流转，根据各组件特点及需求，任务在其中按三种方式流转。

在MySQL中，任务储存以自增id为主键，包含有商品名、商品ID作为任务搜索信息，评论数作为爬虫结果信息，好评率、词云URL、热点词作为数据分析结果以及状态码作为任务流转状态标示。

文本

描述已自动生成

图二：MySQL中任务表

在Redis中，受限于Redis内存存储，非永久存储的特点以及作为消息队列和任务去重的任务要求，任务仅通过任务ID/商品ID存在于Redis中。

Elastic Search作为搜索引擎，向外提供分词解析及模糊搜索功能，储存有任务ID、商品名及商品ID，据此支持商品名及商品ID的搜索功能。

* + 1. 任务的流转

在本设计中，任务具有如下四种状态：创建、等待、处理中、已完成。当用户通过服务端创建任务后，服务端对应进行创建，将任务储存在MySQL中，标记为创建状态，并同步到Redis及Elastic Search。用户主动或定时提交任务进入消息队列时，会将任务ID注册进入消息队列，并将状态修改为等待态。在爬虫测从消息队列中“消费”得到任务ID后，会将任务状态流转至处理中，并进行爬虫爬取和数据处理。全部流程完成后，将任务标记为已完成。

考虑到任务处理中可能会出现的异常情况和网络超时等特殊情况，任务可能丢失下一步流转的判定点，例如任务在处理中状态下时，爬虫端中断处理，导致任务无法进入数据分析，设计采用了定时服务对异常任务进行重置，返回创建状态重新提交。

图形用户界面, 应用程序

描述已自动生成

图三：任务状态流转

* 1. 接口功能实现
     1. 任务生成

任务生成分为两种情况，具体流程如图四所示。对于用户传入的关键词，后台首先根据京东搜索页面链接拼接规则获取搜索结果页面。根据多次搜索经验发现，搜索页前8项商品多为销量高、评论多的京东自营商品，更能表现出商品的真实评价。选定商品后根据商品详情页链接规则获取商品详情，得到商品名及商品ID。

获取ID后，查询Redis Set，判断该商品ID是否曾经已生成，存在于数据库。若已存在，则不再重复进行生成，以避免不必要的重复任务。

确认任务需要生成后，将任务保存进入MySQL，此时会为其分配唯一的任务ID。保存成功后再将该任务的商品ID写入Redis Set。该保存顺序是为了防止MySQL保存失败时商品ID已经写入Redis Set的情况出现，该情况会导致异常情况阻塞重新生成任务。

之后将商品名、任务ID和商品ID组合存入Elastic Search。该组件配置mapping时使用ik\_max\_word分词器，支持将商品ID分解为多个词组保存。在搜索时，根据词语间匹配计算相似度来进行模糊搜索。

图示

描述已自动生成

图四： 任务生成流程

* + 1. 任务提交

对于任务的提交，服务端提供用户主动提交接口及定时自动提交功能。其中自动提交功能作为协程运行，每小时自动调用进行检查。

任务提交首先从MySQL中查询所有状态是创建的任务，若未找到则结束。对于每个任务，将其状态更新至等待，并将其任务ID注册进消息队列，等候爬虫端获取。

* + 1. 任务查询

任务查询主要由Elastic Search支持。根据用户传入的参数，如果是id参数，则进行精确匹配，如果是关键字，则进行模糊匹配，取相似度超过预设阈值的结果。将所有的命中结果任务ID向MySQL进行批量查询，得到所需结果。

文本

描述已自动生成

图五：Elastic Search搜索配置代码

* + 1. 异常任务监测重置

异常任务的监测作为定时任务，每小时自动运行。

对于状态流转至运行中和已完成的任务，若其爬取评论数小于所预期的阈值，或未能获取到好评率，则视为异常任务。若任务阻塞在运行中，则将其和对应的已获取评论删除，重新生成具有相同商品ID的新任务进行处理。若任务处于已完成状态，则认为是商品状态异常，评论较少，如预售状态和广告商品。此时直接删除任务，不重新生成处理。

1. 爬虫的设计及实现
   1. 任务监听及多机协同处理

考虑到任务生成时间开销与爬虫完成任务时间开销较大的差距，本设计采用任务监听机制对生成的任务进行分配，交由爬虫端处理。设计采用Redis List构成作为轻量级消息队列，采用左进右出的方向，将生成的任务ID 排列进消息队列等待处理。爬虫端可采取多机协同的方式，在多台机器上启动并进行监听。通过python-redis支持的brpop阻塞式获取队列消息并处理，通过这种方式来提高整个系统的处理能力。

手机屏幕的截图

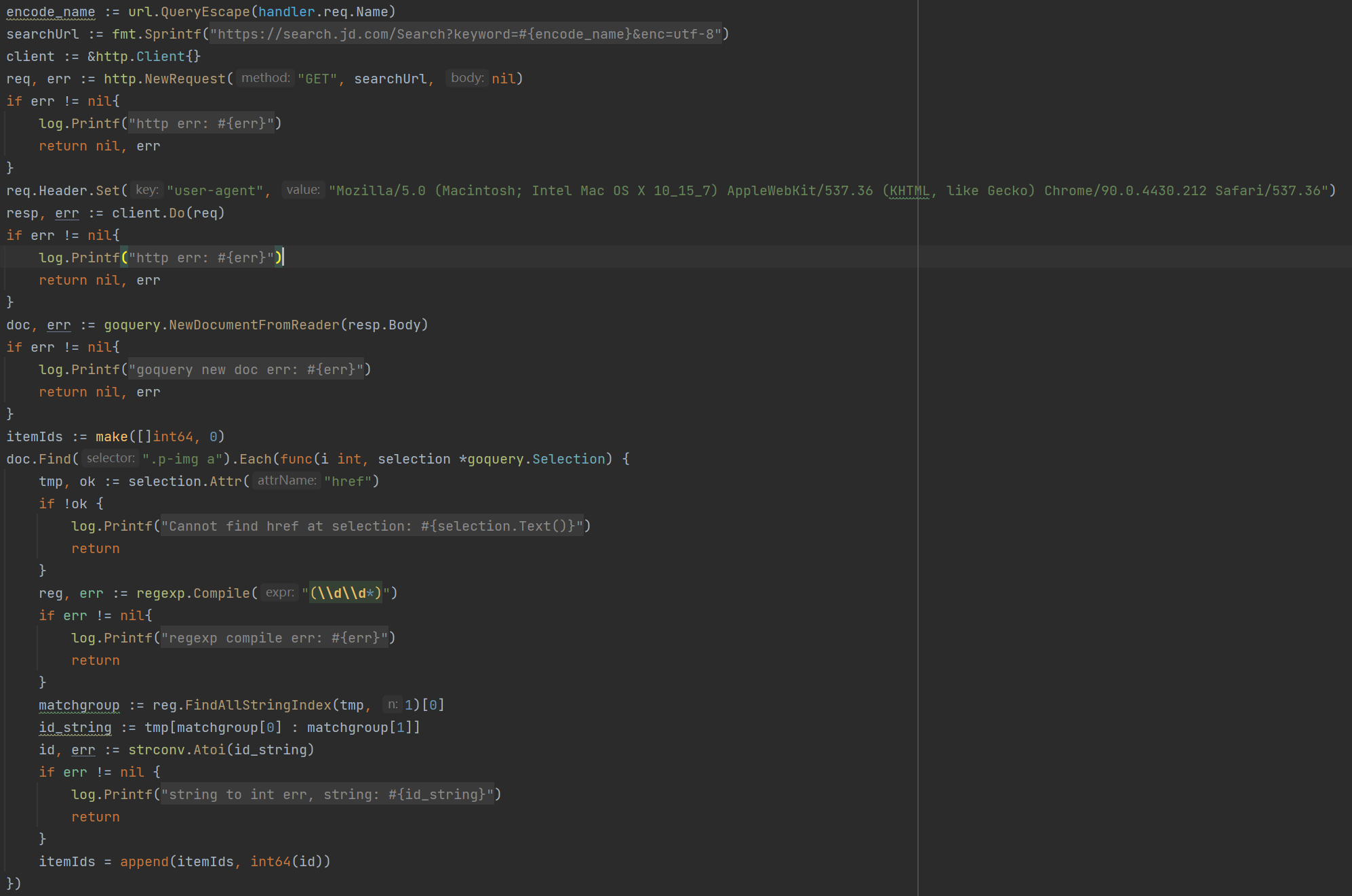
描述已自动生成

图六： 多爬虫端监听消息队列

* 1. 爬取原理及API接口解析

爬虫是指通过向特定URL发送请求，获取网页或数据，并对数据进行解析的技术。在该设计中，共涉及到三个URL的信息爬取。

第一，商品搜索页信息爬取。在生成任务时，需根据用户输入的关键词进行搜索，得到所需的商品列表。根据观察，京东网站的搜索页面需向网址[https://search.jd.com/Search]发送GET请求，并附带keyword和enc参数。其中keyword参数即搜索关键词，需对中文进行URL编; enc参数为结果编码方式，固定传入utf-8进行编码。例如，对于关键词“华为P40”，请求URL即[https://search.jd.com/Search?keyword=%E5%8D%8E%E4%B8%BAP40&enc=utf-8]。据此可得到搜索结果网页HTML文本。此后，通过goquery库进行HTML解析，由CSS选择器语法找到对应商品ID，交由下一步爬取。



图七：商品搜索页信息爬取代码

第二，商品名获取。该部分需要通过商品ID爬取商品详情页获取商品名。商品详情页的URL地址为[https://item.jd.com/67126015776.html]， 其中[67126015776]部分为给定的商品ID，通过向该URL发送GET请求获取商品详情页HTML代码，并通过解析得到对应商品名。

第三，评论获取。根据观察，京东向外提供评论获取接口API，无需进行Cookies登陆校验，可通过配置参数发送GET请求得到相应商品的评论结果，相应参数列表参见表一。

|  |  |
| --- | --- |
| ProductId | 商品ID |
| Score | 好评/中评/差评筛选项 |
| sortType | 推荐排序/时间排序筛选项 |
| Page | 分页页面 |
| Pagesize | 每页评论数，最高为10 |

表一： 评论获取参数表

通过该URL，可获得Json文本串，将其进行解析后发现，在“comments”键下所属的列表即为评论相关信息，将其摘选保存。

* 1. 代理池处理

代理池是指一系列自动更新的代理IP列表。这里采集了互联网上提供的免费代理IP列表。考虑到评论获取所需HTTPS协议请求，我们发现采集的代理IP不可用率达到90%以上，且存活时间极短，响应时间极慢，故仅在代码中留有代理接入开关，未实际采用。

对于爬取的代理IP列表，需要通过向网站[https://ip.cn]发送GET请求进行可通性测试，通过测试的代理IP进行临时存储，并提供爬虫使用。当发现选择的代理IP不通时，需将其剔除，避免超时。

1. 数据分析系统
   1. 数据清洗

定义一个函数，功能是对输入文件中的重复数据进行剔除，然后把去重后的内容输出到新文件中。

文本

描述已自动生成

图八： 去重相关代码

* 1. 自然语言处理

调用SnowNLP做情感鉴别，设定情感极性值的区分值为0.1，超过0.1的为好评，小于0.1的为差评，这一步骤之后商品评论会被分为好评和差评两个评论文件。下图代码为筛选差评的代码：

文本

描述已自动生成

图九： 差评筛选相关代码

* 1. 分词

使用jieba库对评论做分词处理，其中还会去除一些无意义的字段以及筛选出形容词。

文本

描述已自动生成

图十： jieba分词处理

3.4 词频统计

使用Counter做统计，然后筛选好评和差评中的热词，生成json形式的数据以便存入MySQL。

图表, 散点图

描述已自动生成

图十一： Counter统计

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图十二： json生成

* 1. 生成词云

调用WordCloud根据分词处理后的商品评论生成词云。

文本

描述已自动生成

图十三： 词云生成

四、结果分析

对研究过程中所获得的主要的数据、现象进行定性或定量分析，得出结论和推论。

（正文格式：宋体，小4号，不加粗，两端对齐，行距为固定值20磅）

五、课程设计总结

在本次课程设计实验中，我们完成了本次的5G手机评价分析项目。此次项目中，通过实践我们对于软件工程有了更加深入的理解。首先是项目需求的重要性：项目需求是整个项目中重要的路标，能否深度，正确理解项目需求对于整个项目是否能够圆满完成是至关重要的。我们在本次项目中因为对需求分析较为细致，才少走了这个弯路。然后是需求分析与项目设计，我们在正式开始做项目之前做了十分细致的规划，首先是将项目需求解析，并将其拆解合成为软件工程中的流程图，整个项目的每个步骤都十分详细，并且有对应到每个人的分工，因此整个项目做下来目标明确且高效。

其次，我们也学习了不少相关的知识并加以实践和应用：我们学习了使用python设计爬虫，并学习实践了爬虫中的反针对设计，并且在实现爬虫项目中学习使用了redis，从中实现模拟消息队列，提升了爬虫的效率。我们还学习了mysql的相关知识，将其用于了数据库的实现以及数据库与项目的各种交互操作。在文本解析方面，我们学习掌握了python中的jieba组件并在分词中进行使用，实践证明jieba组件是一种使用简单而高效的分词组件，整个评价文本处理部分都依赖于jieba组件来实现。

最后，本次项目中我们也遇到了不少困难，项目刚开始的第一个困难就是各个购物网站的爬虫，由于目前各大购物网站都有相应的反爬虫措施，并导致我们在初次尝试爬虫时遭到IP封禁。因此我们特别地对爬虫部分进行专门设计，以提高爬虫效率。其次，在全组整合项目的过程中，即使在分工明确的情况下，在合并中也出现了些许bug，使得我们必须花费额外的时间来解决。