基于 Python 爬虫的 B 站视频弹幕情感分析

专　　业:　人工智能

班　　级:　B20 智能二班

姓　　名:　潘霞

学　　号:　20086052032

项目名称	基于 Python 爬虫的 B 站视频弹幕情感分析			
项目成员	姓名	学号	分工及贡献	成绩
	潘霞	20086052032	负责代码开发、文档修改整理、PPT 制作及项目书第三章(主要代码和实施过程)的编写。	
	张兴窕	20086052034	负责代码开发及项目书第五章（小组项目介绍、项目总结与贡献、学习体会）的编写。	
	赖成愿	20086052027	负责代码开发以及项目书的第一章(项目背景、概述、目标)的编写。	
	简光婵	20086052033	负责代码开发、PPT 视频前三点的讲解及项目书第二章(项目实施方案介绍)的编写。	
	柯西龙	20086052053	负责代码开发、PPT 视频后两点的讲解及项目书第四章(项目展示及测试)的编写。	
指导老师	海涛			

目 录

1 项目背景、概述及目标

1.1 项目背景

随着互联网的发展，视频弹幕成为了越来越多年轻人在观看视频时的主要交流形式。弹幕[1]文本具有实时性、个性化、情感化等特点，对于理解用户观看视频时的情感、兴趣爱好以及社交行为具有重要意义。因此，对视频弹幕进行爬虫和情感分析的项目显得尤为重要。

首先，通过爬取视频弹幕[2]，我们可以获取关于用户对于该视频的看法、感受、评价等信息，以及他们之间的交流和互动。同时，爬取大量视频弹幕还可以为其他数据挖掘和机器学习任务提供大量有价值的数据样本，如信息检索、用户画像等。

其次，针对观众情感的分析可以为视频制作者提供更为精准的反馈，以帮助他们更好地解读观众的需求和反应，从而改善视频内容和提高用户体验。情感分析[2]还可以在广告营销、舆情监控等领域发挥重要作用。例如，一些商家可以通过分析用户评论等社交媒体数据，快速了解消费者对其产品或服务的评价和反馈，以更好地制定营销策略和推广方案。

最后，这样的项目也为互联网监管提供了有力的手段。一些不良信息和影响公共安全的行为常常可以通过社交媒体用户的交流和信息流中发现。对于这些信息的及时发现和分析可以帮助互联网监管机构及时采取措施，保护公共利益和社会安全。

对视频中弹幕的爬虫是一种用于采集网络直播、视频或其他流媒体网站中用户发送的弹幕信息的工具。其背景主要是由于当前网络直播和视频平台已经成为人们娱乐和获取信息、知识的主要来源之一。在这些平台上，弹幕[2]

已成为用户互动交流的主要方式，用户可以通过弹幕发送各种信息，如提问、评论、表情等。而对于网站的管理者，弹幕也是了解用户需求和对节目的反馈的重要途径。因此，从弹幕中获取有价值的信息，对于推进互联网信息采集与分析具有重要意义。

弹幕是 B 站（Bilibili）视频中用户自发生成的短文本，通常表达着用户对视频内容的反馈和情感，可直观、快速、有效地反映用户在看视频时的情感和心态。对这些弹幕进行情感分析可以帮助我们更好地了解用户对视频的情感态度，并为 B 站（Bilibili）的运营和内容生产提供更多的数据支持。因此，越来越多的研究者和数据分析师们开始了对 B 站（Bilibili）视频弹幕情感分析的研究。基于此背景，我们学习小组投入到弹幕情感分析的学习和研究中。

1.2 项目概述

基于 python 爬虫[3]的王心凌 B 站视频弹幕情感分析，该项目旨在使用 Python 编写一个爬虫，从视频分享网站获取王心凌的视频弹幕数据。其次，对弹幕进行情感分析。

具体来说，项目的步骤如下：

(1) 使用 Python 编写爬虫[4]，从视频分享网站获取包含王心凌的视频弹幕数据。使用 Python 的 requests 库向 B 站的弹幕 API 发送请求，获取视频的弹幕数据。

(2) 使用 HTML 解析器[4]库 BeautifulSoup 对获取到的网页 HTML 源码进行解析，提取弹幕文本信息。对获取的弹幕数据进行数据清洗和预处理，包括去除重复弹幕、过滤掉无关信息等。

(3) 对采集得到的弹幕情感数据进行去重、过滤，清理不必要的符号和字

样，以保证数据的准确性和可靠性。 使用 jieba 进行分词，使用 SnowNLP 对分词过后的文本进行情感分析，得到各个弹幕的情感极性，将其归为“正向情感”、“负向情感”和“中性情感”三类。

(4) 使用 jieba 进行分词，使用 SnowNLP 对分词过后的文本进行情感分析[5]，得到各个弹幕的情感极性，将其归为“正向情感”、“负向情感”和“中性情感”三类。并用 jieba 统计 Top10 高频词

(5) 通过可视化的方式，将情感分析的结果展示出来，使用 matplotlib 绘制饼状图来展示弹幕情感极性的占比。使用 wordcloud 库绘制词云图，展示弹幕中出现频率较高的关键词。

(6) 可以加入一些控制因素，来进一步分析弹幕的情感波动。例如按发布时间划分的小时、天等。

总之，该项目旨在使用 Python 爬虫技术和自然语言处理技术，对视频弹幕数据进行情感分析，以更好地了解观众对王心凌的表现的反应和情感。

1.3 项目目标

对视频中弹幕进行爬虫[4]和情感分析的项目的主要目标是帮助视频内容生产者或平台运营者更好地了解用户对其视频内容的反馈和情感态度，以便于针对用户需求进行改进和优化。

具体来说，本项目的目标包括以下四个方面：

(1) 收集视频弹幕数据：通过网络爬虫技术获取视频弹幕数据[4]，包括文本内容、发送时间、用户 id 等信息。

(2) 数据预处理：对收集到的数据进行处理和清洗，例如去除重复数据、过滤不相关的信息、将文本进行分词等。

(3) 情感分析[5]：使用 jieba 进行分词及统计 Top10 高频词，使用 SnowNLP

对分词过后的文本进行情感分析，根据用户文本的情感色彩对弹幕进行分类汇总，并提供词云或趋势图等形式的可视化展现。

(4) 提供决策支持：通过对用户反馈和情感的分析，提供给视频内容生产者或平台运营者决策支持，帮助他们更好地了解用户需求和偏好，优化视频内容，提高用户体验。

综上所述，通过对视频中弹幕进行爬虫和情感分析[5]，可以让视频内容生产者或平台运营者更好地了解用户需求和反馈，提高视频内容质量和平台用户体验。

2 项目实施方案

2.1 相关算法

2.1.1 分词算法

jieba 库是 Python 中的优秀的中文分词第三方库。可以用来进行关键字搜索，中文文本需要通过分词获得单个的词语，jieba 库最强大的功能之一就是对爬取的弹幕中出现的词汇进行计数统计，即计算高词频，高频词是指文档中出现频率较高且非无用的词语，其一定程度上代表了文档的焦点所在。针对单篇文档可以作为一种关键词来看。对于如新闻这样的多篇文档，可以将其作为热词，发现舆论热点。对于一篇文章或者弹幕爬虫，我们可以通过以下步骤对出现的单词进行统计，如下图 2.1：

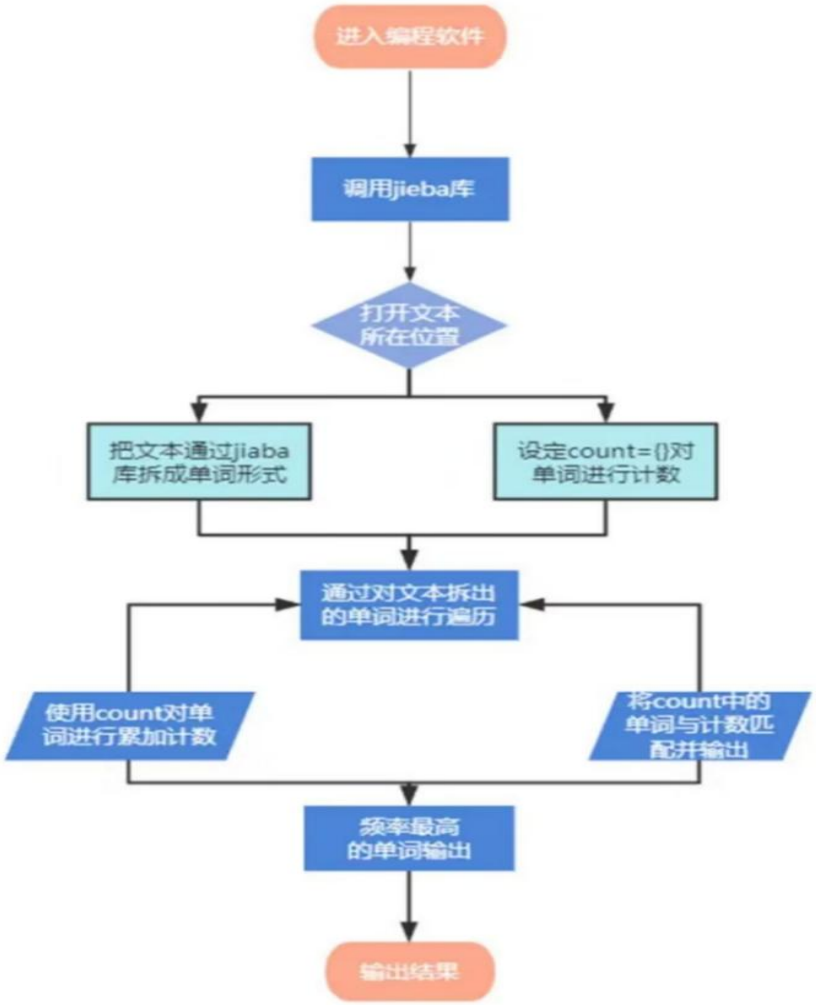

图 2.1 jieba 库进行高频统计步骤图

2.1.2 情感分析算法

情感分析算法是一种人工智能技术，用于分析文本、语音或图像数据中的情感状态和倾向性。它被广泛用于市场调研、社交媒体和客户服务等领域，以帮助企业了解客户需求和情感偏好，优化营销策略和提高客户满意度。情感分析算法的核心原理是使用自然语言处理模型，将文本数据转换为“情感”和“情感倾向”两个维度的数据，其中情感指人的情绪状态，如喜、怒、哀、乐等；情感倾向指人的行为倾向或态度，如支持、反对、中立等。使用 SnowNLP、TextBlob 等库对文本进行情感判定，判断结果归为“正向情感”、“负向情感”和“中性情感”三类。

2.1.3 异常词过滤算法

异常词过滤算法是使用自定义“stop words”列表过滤无意义或异常词，如标点符号、停用词等。本项目中使用 WordCloud 词云图去除停用词的正确方法用 wordcloud 库制作中文词云图，必须要分词，在分词前，在 wordcloud 中设置 stopwords。

2.1.4 可视化算法

数据可视化是指将数据放在可视环境中、进一步理解数据的技术，可以通过它更加详细地了解隐藏在数据表面之下的模式、趋势和相关性。 使用 matplotlib 库进行数据可视化，Matplotlib 是一个在 python 下实现的类 matlab 的纯 python 的第三方库,旨在用 python 实现 matlab 的功能,matplotlib 对于图像美化方面比较完善，可以自定义线条的颜色和样式，可以在一张绘图纸上绘制多张小图，也可以在一张图上绘制多条线，可以很方便地将数据可视化并对比分析。

Matplotlib 模块依赖于 NumPy 和 tkinter 模块，可以绘制多种形式的图形，包括线图、直方图、饼图、散点图等，图形质量满足出版要求，是数据可视化的重要工具。本项目中使用 matplotlib 库进行数据可视化。绘制饼状图和词云图等进行展示。

2.2 技术栈的介绍及使用

情感分析的第一步是获取数据，而网络尤其是社交网络是存在着丰富而易于获得的意见型数据资源。 选择爬虫种类(Spider, CrawlSpider), 取决于目标和爬虫各自合适的应用场景，有一个能够生成请求(request)的“解析(parse)”方法。基于 Python 爬虫的网站视频弹幕进行情感分析项目中使用的技术栈有：

Pandas 技术栈、jieba 分词技术栈、爬虫抓取技术栈。

2.2.1 Pandas 技术栈

Pandas 是 Python 的核心数据分析支持库，提供了快速、灵活、明确的数据结构，旨在简单、直观地处理关系型、标记型数据。它建立在 NumPy 之上，使得处理结构化数据更加简单和高效。Pandas 的两个主要数据结构是 Series 和 DataFrame，可以理解为 NumPy 数组的增强版。它们提供了更多的功能和灵活性，使得数据处理变得更加直观和方便。

2.2.2 爬虫抓取技术栈

网络爬取是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本，通过程序去获取 web 页面上自己想要的数据，也就是自动抓取数据。拟浏览器打开网页，获取网页中我们想要的那部分数据，浏览器打开网页的过程：当你在浏览器中输入地址后，经过 DNS 服务器找到服务器主机，向服务器发送一个请求，服务器经过解析后发送给用户浏览器结果，包括 html,js,css 等文件内容，浏览器解析出来最后呈现给用户在浏览器上看到的结果，所以用户看到的浏览器的结果就是由 HTML 代码构成的，我们爬虫就是为了获取这些内容，通过分析和过滤 html 代码，从中获取我们想要资源。爬虫的基本流程如下：

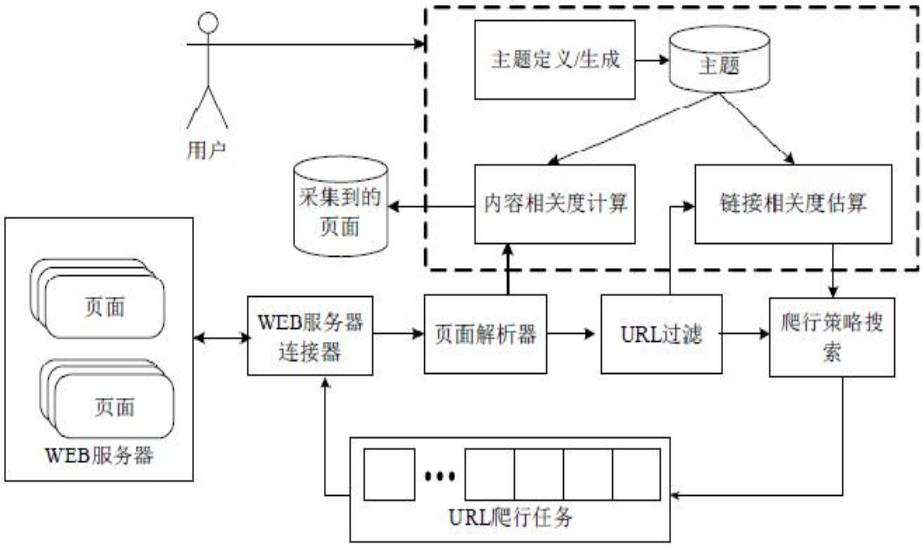

图 2.2 爬虫流程图

2.2.3 jieba 库技术栈

由于中文文本中的单词不是通过空格或者标点符号分割，所以中文及类似语言存在一个重要的“分词” 问题，jieba、SnowNLP(MIT)、pynlpir 等都可以完成对中文的分词处理，该文章采用 jieba 进行中文分词。jieba 库是一个进行中文分词的第三方库。可用来进行关键字搜索。jieba 是一个 python 实现的分词库，对中文有着很强大的分词能力，我们在使用时通过 import jieba 导入 jieba 库：中文文本需要通过分词获得单个的词语；jieba 分词依靠中文词库：利用一个中文词库，确定汉字之间的关联概率；汉字间概率大的组成词组，形成分词结果；除了分词，用户还可以添加自定义的词组。jieba 库有三种分词模式，精确模式：试图将句子最精确地切开，适合文本分析(默认是精确模式)；全模式：把句子中所有的可以成词的词语都扫描出来，速度非常快，有冗余，不能解决歧义；搜索引擎模式：在精确模式的基础上，对长词再次切分，适合用于搜索引擎分词。利用 jieba 技术栈统计 Top10 个高频词，用与分词，停用词绘制云图。

3 主要代码及实施过程

3.1 主要代码

3.1.1 数据爬取模块

(1) 相关库的导入

```
import re
import requests
from bs4 import BeautifulSoup as BS
import time
import pandas as pd
import os
```

导入 re、requests、BeautifulSoup、time、panda、os 等库用于正则表达式提取文本、向网站发送爬虫请求、解析 HTML/XML 页面和处理数据并存储为 CSV 文件。

(2)定义 get_bilibili_danmu()函数，用于获取 B 站弹幕并保存到 CSV 文件

① 向视频地址发送请求，解析出 cid 号

```
def get_bilibili_danmu(v_url, v_result_file):
    headers = {'User-Agent': "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)", }
    print('视频地址是：', v_url)
    r1 = requests.get(url='https://api.bilibili.com/x/player/pagelist?bvid='+bv, headers=headers)
    html1 = r1.json()
    cid = html1['data'][0]['cid']
    print('该视频的cid是:', cid)
```

定义字典 headers 请求 B 站的网页，客户端使用 Mozilla Firefox 浏览器，使用 requests 模块向 Bilibili 的 API 地址发送 GET 请求，请求参数为 bv，响应结果按 JSON 格式返回，以 r1 对象保存，r1 对象包括了相应网页的状态码、响应头和响应体等信息，解析后的结果使用字典语法访问和处理，调用 json() 方法，将响应体解析为一个 Python 字典，保存在 html 变量中，获取含有视频信息的字典 html，然后通过字典索引方式（即使用键索引值）获取到该视频对应的 cid。

② 根据 cid 号，拼接 xml 接口地址，再次发送请求

```
17      danmu_url = 'http://comment.bilibili.com/{}.xml'.format(cid)
18      print('弹幕地址是：', danmu_url)
19      r2 = requests.get(danmu_url)
20      html2 = r2.text.encode('raw_unicode_escape')
```

根据 cid 号生成弹幕地址（danmu_url），保存在变量 danmu_url 中，并打印出来，发送请求获取弹幕信息，保存在变量 r2 中。由于获取到的响应内容为字节串，需要使用 encode 方法将其转换为字符串，并采用 raw_unicode_escape 编码方式。

③ 解析 XML 页面

```
21      soup = BS(html2, 'xml')
22      danmu_list = soup.find_all('d')
23      print('共爬取到{}条弹幕'.format(len(danmu_list)))
24      video_url_list = []
25      danmu_url_list = []
26      time_list = []
27      text_list = []
28      for d in danmu_list:
29          data_split = d['p'].split(',')
30          temp_time = time.localtime(int(data_split[4]))
31          danmu_time = time.strftime("%Y-%m-%d %H:%M:%S", temp_time)
32          video_url_list.append(v_url)
33          danmu_url_list.append(danmu_url)
34          time_list.append(danmu_time)
35          text_list.append(d.text)
36          print('{}:{}'.format(danmu_time, d.text))
```

使用 BeautifulSoup 库解析弹幕网页的 XML 格式内容，找到所有的弹幕标签（d 标签），将其中的弹幕时间、弹幕内容等信息保存在不同的列表中，并输出到控制台上。

④ 为避免多次写入 csv 标题头，加上逻辑处理

```
    df = pd.DataFrame()
    df['视频地址'] = video_url_list
    df['弹幕地址'] = danmu_url_list
    df['弹幕时间'] = time_list
    df['弹幕内容'] = text_list
    if os.path.exists(v_result_file):
        header = None
    else:
        header = ['视频地址', '弹幕地址', '弹幕时间', '弹幕内容']
    df.to_csv(v_result_file, encoding='utf_8_sig', mode='a+', index=False, header=header)
```

初始化一个 DataFrame 对象，将列表中的各项数据整合在一起，使用 pandas 库生成 DataFrame，将其写入到结果文件中，以逗号分隔的 CSV 格式写入，字符编码为 UTF-8 并且没有 BOM 头。如果结果文件已经存在，则采用文件追加模式，否则需要写入表头。如果文件存在，不需写入字段标题，如果文件不存在，说明是第一次新建文件，需写入字段标题，并将数据保存到 CSV 文件。

3.1.2 情感分析模块

(1) 相关库的导入

```
import pandas as pd
from snownlp import SnowNLP
from wordcloud import WordCloud
from pprint import pprint
import jieba.analyse
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
```

导入 pandas、SnowNLP、WordCloud、pprint、jieba 等库，用于数据分析，进行中文情感分析，绘制词云图，美观打印，分词，读取图片并将图片的像素点转换成矩阵数据。

(2) 解决中文显示问题

```
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
```

显示中文标签,指定默认字体，解决保存图像是负号'-'显示为方块的问题。

(3) 弹幕情感分析打分

```
def sentiment_analyse(v_cmt_list):
    score_list = []
    tag_list = []
    pos_count = 0
    neg_count = 0
    mid_count = 0
    for comment in v_cmt_list:
        tag = ''
        sentiments_score = SnowNLP(comment).sentiments
        if sentiments_score < 0.5:
            tag = '消极'
            neg_count += 1
        elif sentiments_score > 0.5:
            tag = '积极'
            pos_count += 1
        else:
            tag = '中性'
            mid_count += 1
        score_list.append(sentiments_score)
        tag_list.append(tag)
    df['情感得分'] = score_list
    df['分析结果'] = tag_list
```

定义 sentiment_analyse()函数，使用 SnowNLP 库对每条评论进行情感分析，并根据得分将其分类为积极、中性或消极。情感得分是一个 0 到 1 之间的数，越接近 1 表示越积极，越接近 0 表示越消极，等于 0.5 则表示中性情感。使用列表分别记录每条评论的情感得分和分类结果，最终将其作为两列添加到一个名为 df 的 DataFrame 对象（函数中未给出 df 的定义）中。该函数还记录了每种分类下的评论数量，并将其分别计入了变量 pos_count、neg_count 和 mid_count 中。

(4) 绘制占比图

```
grp = df['分析结果'].value_counts()
print('正负面评论统计：')
print(grp)
grp.plot.pie(y='分析结果', autopct='%.2f%%')
plt.title('王心凌弹幕_情感分布占比图')
plt.savefig('王心凌弹幕_情感分布占比图.png')
plt.show()

df.to_excel('王心凌弹幕_情感评分结果.xlsx', index=None)
print('情感分析结果已生成：王心凌_情感评分结果.xlsx')
```

对进行情感分析后得到的分类结果进行统计，并将结果以饼图形式展示。其中，grp 变量使用 pandas 库中的 value_counts()方法对分类结果进行统计，得到每种情感分类下的评论数量，并作为一个 Series 对象存储在 grp 中。函数中使用 print()函数输出了统计结果。接着，使用 matplotlib 库的 plot 和 savefig 方法生成并保存饼图。这里采用了.plot.pie()方法，同时指定 y 轴为‘分析结果’列，autopct 参数控制饼图上区块显示的格式，%.2f%%表示保留两位小数的百分比形式。图像保存在当前目录下的‘王心凌弹幕_情感分布占比图.png’文件中。最后，使用 pandas 库的 to_excel 方法将 DataFrame 对象中的情感得分和分类结果保存至名为‘王心凌弹幕_情感评分结果.xlsx’的 Excel 文件中，并再次使用 print()函数输出保存成功的信息。

(5) 绘制词云图

```python
def make_wordcloud(v_str, v_stopwords, v_outfile):
    print('开始生成词云图：{}'.format(v_outfile))
    try:
        stopwords = v_stopwords
        backgroud_Image = np.array(Image.open('王心凌_背景图.png'))
        wc = WordCloud(
            background_color="white",
            width=1500,
            height=1200,
            max_words=1500,
            font_path="C:\Windows\Fonts\simhei.ttf",
            stopwords=stopwords,
            mask=backgroud_Image,
        )
        jieba_text = " ".join(jieba.lcut(v_str))
        wc.generate_from_text(jieba_text)
        wc.to_file('王心凌弹幕_词云图.png')
        plt.imshow(wc, interpolation='bilinear')
        plt.axis("off")
        plt.show()
        print('词云图生成成功：王心凌弹幕_词云图.png')

    except Exception as e:
        print('make_wordcloud except: {}'.format(str(e)))
```

定义 make_wordcloud()函数，使用 WordCloud 库生成基于指定文本的词云图，并将图像保存在‘王心凌弹幕_词云图.png’文件中。

具体来看，该函数接受三个参数：v_str 表示需要生成词云图的文本，v_stopwords 表示需要过滤的停用词列表，v_outfile 为保存词云图的文件名。函数中使用了 numpy、jieba 和 WordCloud 库。主要功能由 WordCloud 库实现，由于词云图常常需要根据一张背景图片来生成，函数中使用 np.array(Image.open(‘王心凌_背景图.png’))打开了一张命名为‘王心凌_背景图.png’的背景图片，并将其转化为 numpy 数组。接着，配置了 WordCloud 对象的参数，如背景色、画布大小、最大词数、字体、停用词等，并调用.generate_from_text()方法生成词云图。将生成的词云图使用.to_file()方法保

存至本地。最后，展示并打印词云图生成成功的信息。

(6) 用 jieba 统计 top 10 高频词

```
keywords_top10 = jieba.analyse.extract_tags(v_cmt_str, withWeight=True, topK=10)
print('top10关键词及权重：')
pprint(keywords_top10)
with open('TOP10高频词.txt', 'w') as f:
    for i in keywords_top10:
        f.write(str(i[0]) + ' ' + str(i[1]))
        f.write('\n')
```

使用 jieba 库提取指定文本的关键词和权重，并将结果输出到控制台和本地文件中。具体来看，该函数接受三个参数：v_cmt_str 表示需要提取关键词的文本，withWeight 表示是否返回词语的权重值，topK 表示返回的关键词数目。函数在使用 jieba 库中的 extract_tags 方法提取指定文本的关键词和权重时设置 withWeight=True 来同时输出词与权重值，预设为提取出现频度最高的 10 个关键词。使用 Python 内置 pprint 模块输出显示关键词及其权重值；接着使用 Python 内置 open 方法打开一个命名为‘TOP10 高频词.txt’的文件，并使用 for 循环依次将每个关键词和对应的权重值以指定格式写入文件中。

3.2 实施过程

3.2.1 项目实施步骤

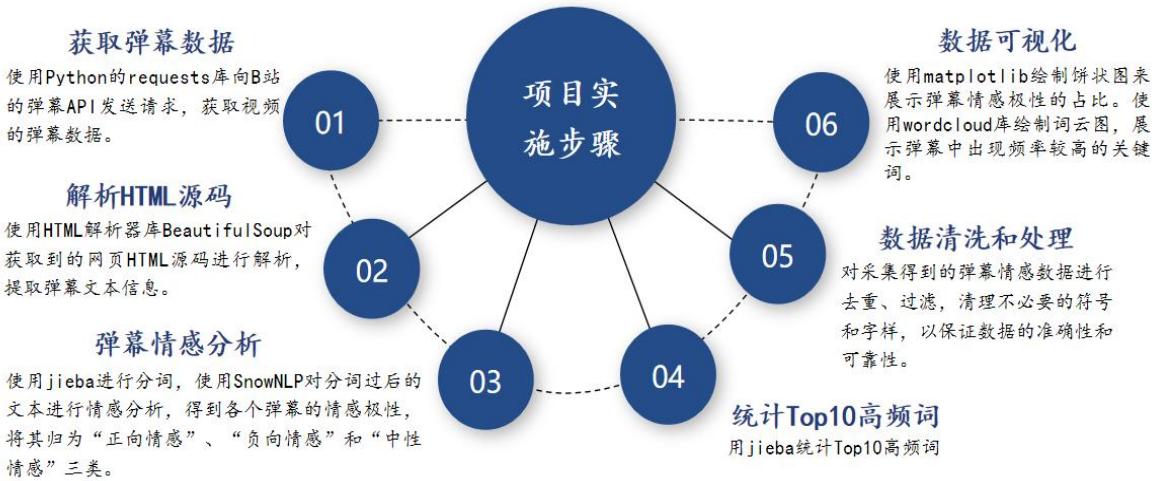

图 3.1 项目实施步骤图

3.2.2 弹幕数据爬取过程

B 站是一家中国视频分享网站，用户可以在其平台观看、分享和上传视频。而弹幕是 B 站的一个特色功能，让用户可以在视频播放器中发表自己的评论，弹幕经过审核后会显示在视频上方，使得用户的评论可以更加直观地体现出来。对 B 站视频的弹幕进行爬取可以分为以下几个步骤：

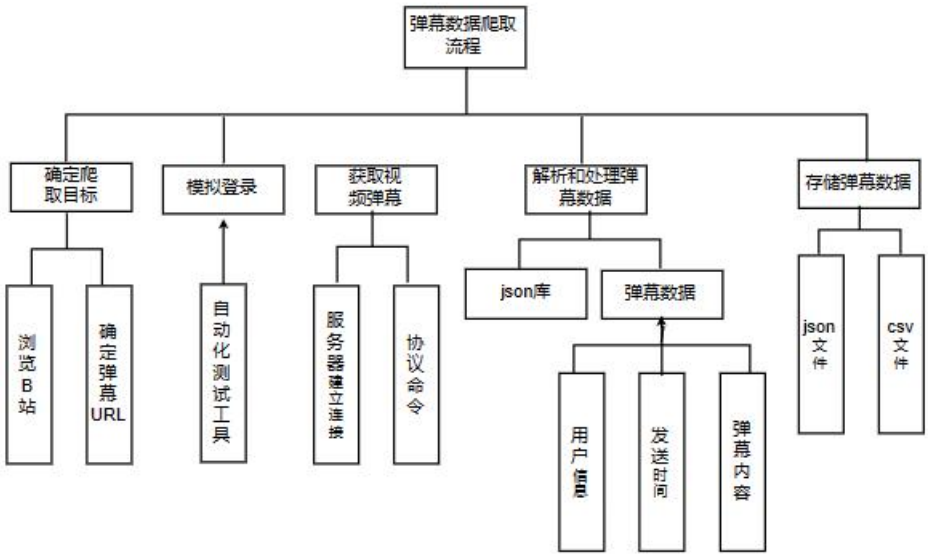

图 3.2 弹幕数据爬取流程图

(1) 确定爬取目标

通过浏览 B 站网站，找到自己感兴趣的视频，并确定需要爬取其弹幕的具体 URL。

(2) 模拟登录

爬取 B 站的弹幕，首先需要模拟登陆 B 站。这可以使用 Selenium 等自动化测试工具模拟人的行为进行。

(3) 获取视频弹幕

使用浏览器工具审查 B 站视频页面后可以发现，视频弹幕是通过 WebSocket 协议实时传输的。因此，我们需要模拟 WebSocket 协议来获取弹幕，

在 Python 中 websocket 和 websocket-client 库都可以用来创建 WebSocket 连接。可以注册 URI wss://broadcastlv.chat.bilibili.com:2245/sub 来与 B 站 弹幕服务器建立 WebSocket 连接，然后使用协议中的相应命令来获取和解析弹幕数据。弹幕数据格式可以在 B 站的开放 API 文档中找到。

(4) 解析和处理弹幕数据

使用 Python 中的 json 库可以将获取到的弹幕数据解析成 Python 对象，并进行处理和分析。弹幕数据包括了发送弹幕的用户信息、发送时间、弹幕内容等信息。

(5) 存储弹幕数据

可以选择不同的方法存储处理后的数据，常见的方式是将数据保存成 JSON 或 CSV 文件并存储到数据库中等。

3.2.3 弹幕情感分析过程

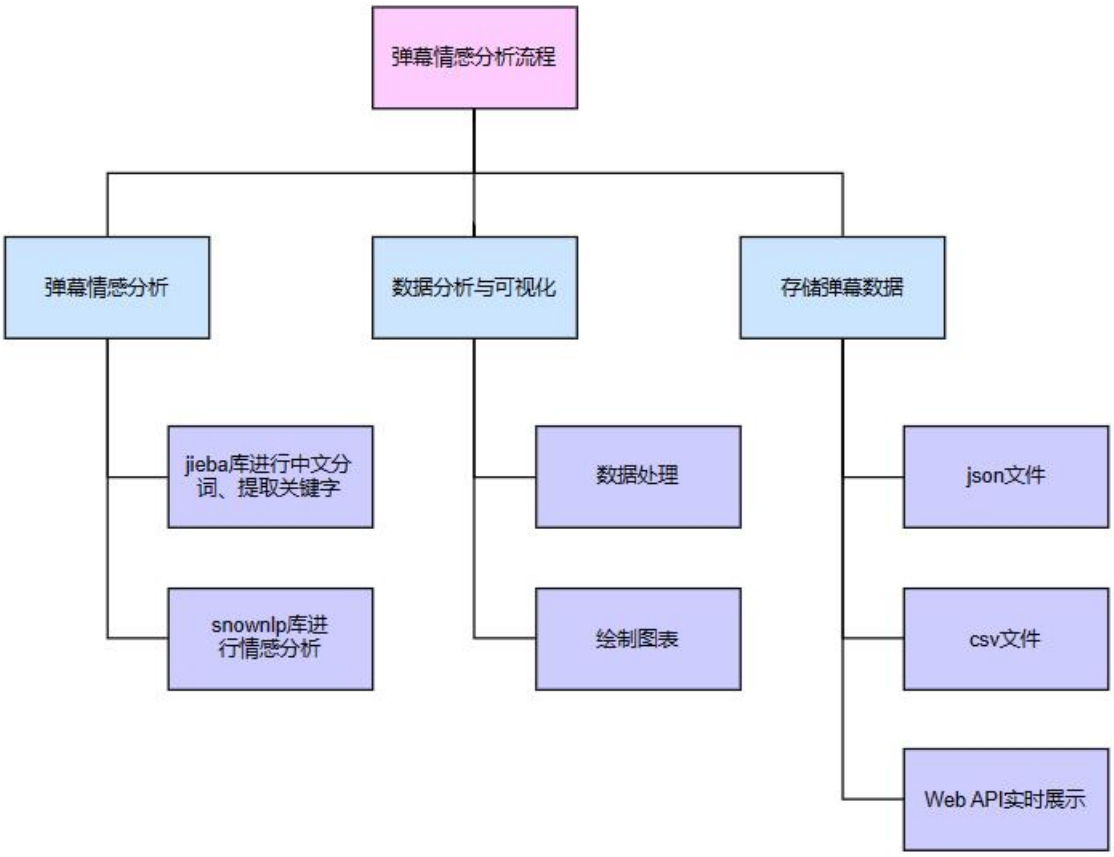

图 3.3 弹幕情感分析流程图

(1) 弹幕情感分析

使用第三方自然语言处理库（如 jieba、snownlp 等）进行文字分词及情感分析。jieba 库可以进行中文分词，提取文本中的关键词；snownlp 库则可以进行情感分析，得出文本的情感分数。使用情感分析结果对弹幕进行情感判别和评分。

(2) 数据分析与可视化

使用各种 Python 数据分析库进行数据处理、绘制图表并展示弹幕情感分析结果。具体可以使用 matplotlib, wordcloud 等库绘制折线图，柱状图、词云图等，以便更好地向用户展示弹幕情感分析的结果。

(3) 存储弹幕数据

选择不同的方式存储弹幕情感分析结果。可以将结果保存成 JSON 文件、CSV 文件，存储到数据库中等。此外，可以使用某些 Python web 框架将结果通过 Web API 向用户实时展示。

4 项目展示及测试

4.1 测试结果 1

```
D:\Anaconda\python.exe "C:/Users/noone/Desktop/Sentiment Analysis Of Wang/1_B站弹幕爬虫.py"
爬虫程序开始执行！
王心凌弹幕.csv已存在，开始删除文件
王心凌弹幕.csv已删除文件
视频地址是： https://www.bilibili.com/video/BV1qY4y157dz
```

这是利用爬虫爬取到 B 站王心凌的相关视频的地址，点击此地址会自动打开浏览器就可以看到 B 站王心凌相关视频。以该地址可以查看网页源代码，就可以找到对应的视频的 cid 是 727777486。所以该视频对应的弹幕接口地址：http://comment.bilibili.com/727777486.xml。

4.2 测试结果 2

弹幕地址是： http://comment.bilibili.com/727777486.xml
共爬取到1200条弹幕

这是利用爬虫爬取到 B 站王心凌的相关视频的弹幕地址，共爬取到了 1200 条弹幕，点击此弹幕地址会打开相关的 XML 文件就可以查看爬取到的全部弹幕的相关内容。这里我截取了部分弹幕，内容如下图：

<chatserver>chat.bilibili.com</chatserver>
<chatid>727777486</chatid>
<mission>0</mission>
<maxlimit>1000</maxlimit>
<state>0</state>
<real_name>0</real_name>
<source>k-v</source>
<d p="33.26300,1,25,16777215,1653919138,0,39460a4c,1063909857352080640,10">04后表示王心凌好甜</d>
<d p="155.02100,5,25,15138834,1653564475,0,b995e636,1060934729404489216,10">焯！！！！！！</d>
<d p="122.72000,1,25,16740868,1653542102,0,1f7f9b05,1060747050347443456,10">除了想笑还有点想哭</d>
<d p="113.16800,1,25,16777215,1653664037,0,1e3e2e3,1061769919139606528,10">为什么我看哭了。。想起了自己的青春5555</d>
<d p="171.98100,5,25,16765698,1653567170,0,d6d43a51,1060957335159149056,10">野 狼 disco</d>
<d p="22.95700,5,25,16646914,1653969118,0,3cf8a41b,1064329121288557312,10">我的胰岛素呐！太甜了吧也</d>
<d p="107.76700,5,25,15138834,1653542901,0,70b78fa1,1060753756335062016,10">变声期</d>
<d p="102.56800,1,25,16777215,1653531384,0,5685cb51,1060657139779049472,10">哈哈哈哈哈哈哈哈哈哈哈哈我笑裂了</d>
<d p="144.78000,1,25,16777215,1653492030,0,5011cca3,1060327020698810112,10">猛男落泪</d>
<d p="16.65900,1,25,15138834,1653321584,0,759c54ed,1058897216346877952,10">通通闪开！大学</d>
<d p="92.76600,1,25,16777215,1653321337,0,91bbb80f,1058895144998172160,10">DNA复苏中</d>
<d p="89.26300,1,25,16777215,1654882750,0,6b569847,1071993222672177664,10">10表示王心凌真的超级甜啊啊啊啊啊啊啊啊</d>
<d p="85.45900,5,25,16707842,1653809950,0,2f54391a,1062993924756555008,10">还是看看书架上的玩偶吧家人们</d>
<d p="9.68100,1,25,16777215,1653648279,0,2bdeb04d,1061637731429515264,10">尴捧</d>

4.3 测试结果 3

2023-02-11 23:22:11:爱你
2023-02-11 23:21:57:爱你
2023-02-11 23:21:43:爱你
2023-02-11 23:21:31:爱你
2023-02-11 23:21:15:爱你
2023-02-11 23:20:57:爱你
2023-02-11 23:20:43:爱你

对于爬取到的弹幕内容，我们可以看到弹幕内容“爱你“是比较多的。反应出王心凌拥有较高的人气。舞台歌曲《爱你》深受粉丝的喜欢，唤起了大部分粉丝们的青春回忆。

4.4 测试结果 4

解析 xml 页面:<d>标签的文本内容为弹幕,<d>标签内 p 属性值.(按逗号分隔)的第四个字段是时间戳。以下是相关代码:

```
soup = BS(html2, 'xml')
danmu_list = soup.find_all('d')
print('共爬取到{}条弹幕'.format(len(danmu_list)))
video_url_list = []
danmu_url_list = []
time_list = []
text_list = []
for d in danmu_list:
data_split = d['p'].split(',')
temp_time = time.localtime(int(data_split[4]))
danmu_time = time.strftime("%Y-%m-%d %H:%M:%S", temp_time)
video_url_list.append(v_url)
danmu_url_list.append(danmu_url)
time_list.append(danmu_time)
text_list.append(d.text)
print('{}:{}'.format(danmu_time, d.text))
```

4.5 测试结果 5

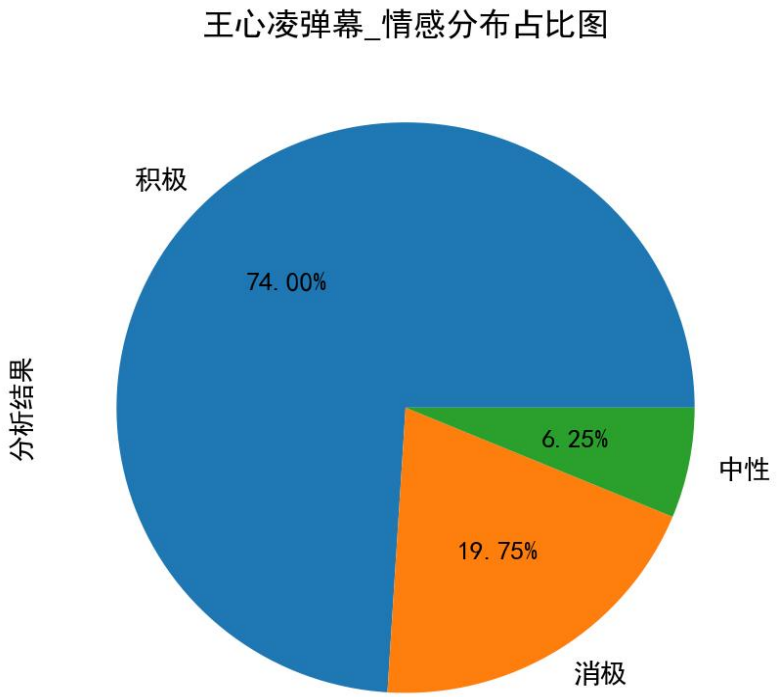

图 4.1 王心凌弹幕情感分析占比图

情感分析计算的得分值、分类打标，并画出饼图。以下是相关代码：

```
for comment in v_cmt_list:
    tag = ''
    sentiments_score = SnowNLP(comment).sentiments
    if sentiments_score < 0.5:
        tag = '消极'
        neg_count += 1
    elif sentiments_score > 0.5:
        tag = '积极'
        pos_count += 1
    else:
        tag = '中性'
        mid_count += 1
    score_list.append(sentiments_score)  # 得分值
    tag_list.append(tag)  # 判定结果
df['情感得分'] = score_list
df['分析结果'] = tag_list
```

从饼图我们可以清晰的看出三种情感的占比。

4.6 测试结果 6

	A	B	C	D	E	F	G
1	视频地址	弹幕地址	弹幕时间	弹幕内容	情感得分	分析结果	
2	https://w	http://con	2022-05-	04后表示	0.987167	积极	
3	https://w	http://con	2022-05-	焯！！！	0.06252	消极	
4	https://w	http://con	2022-05-	除了想笑	0.875806	积极	
5	https://w	http://con	2022-05-	为什么我	0.996525	积极	
6	https://w	http://con	2022-05-	野 狼 di	0.918217	积极	
7	https://w	http://con	2022-05-	我的胰岛	0.586511	积极	
8	https://w	http://con	2022-05-	变声期	0.562582	积极	
9	https://w	http://con	2022-05-	哈哈哈哈	0.996449	积极	
10	https://w	http://con	2022-05-	猛男落泪	0.983405	积极	
11	https://w	http://con	2022-05-	通通闪开	0.557237	积极	

这是弹幕情感评分结果.xlsx.文件的部分截图。以下是将情感分析结果保存到 excel 文件的代码：

```
df.to_excel('王心凌弹幕_情感评分结果.xlsx', index=None)
print('情感分析结果已生成：王心凌_情感评分结果.xlsx')
```

4.7 测试结果 7

```
D:\Anaconda\python.exe "C:/Users/noone/Desktop/Sentiment Analysis Of Wang/2_弹幕的情感分析.py"
length of v_cmt_list is:2400
正负面评论统计：
积极    1776
消极     474
中性     150
Name: 分析结果, dtype: int64
```

这是根据设定情感得分值小于 0.5 为消极，大于 0.5 为积极，等于 0.5 为中性的正负面评论统计结果。共有 2400 条弹幕内容，其中积极的有 1776，消极的有 474，中性的有 150。从中可以看出大部分网友的评论是积极的。积极和中性评价约占 74%，远远大于消极评价。

4.8 测试结果 8

```
心凌王 0.9663727113150881
哈哈哈 0.3680360595551498 4
王心凌 0.24364057764088026
甜心 0.17471705743766056
啊啊啊 0.17320828287389964
王心 0.15950426488850833
哈哈哈哈 0.12454168135699606
教主 0.10926640336221957
心凌 0.10646479022962835
弹幕 0.09001611546802535
```

这是根据全部弹幕统计出的 top10 高频词。代码如下：

```
keywords_top10 = jieba.analyse.extract_tags(v_cmt_str, withWeight=True
topK=10)
print('top10 关键词及权重：')
pprint(keywords_top10)
```

这里需要注意，在调用 jieba.analyse.extract tags 函 数时，要导入的是 import jieba.analyse 而不是 import jieba。频率越高就代表该词在弹幕中出现的次数越多。

4.9 测试结果 9

图 4.2 王心凌弹幕情感词云图

这是根据代码绘制出的词云图其中需要用到 WorldCloud 这个库。需要注意的是想要通过原始图片的形状生成词云图，原始图片一定要白色背景图片，否则生成的是满屏词云！从此词云图可以看出“爱”、“王心凌”、“心凌王”、“哈哈哈”等积极评论较多。

4.10 测试结果 10

图 4.3 王心凌原始与词云对比图

这是原始图片和生成的词云图。我们进行对比可以看到我们的代码生成的词云图与原始图片一致。相关代码如下：

```
try:
    stopwords = v_stopwords
    backgroud_Image = np.array(Image.open('王心凌_背景图.png'))
    wc = WordCloud(
        background_color="white",
        width=1500,
        height=1200,
        max_words=1500,
        font_path="C:\Windows\Fonts\simhei.ttf",
        stopwords=stopwords,
        mask=backgroud_Image,
    )
    jieba_text = " ".join(jieba.lcut(v_str))
    wc.generate_from_text(jieba_text)
    wc.to_file('王心凌弹幕_词云图.png')
    plt.imshow(wc, interpolation='bilinear')
    plt.axis("off")
    plt.show()
    print('词云图生成成功：王心凌弹幕_词云图.png')

except Exception as e:
    print('make_wordcloud except: {}'.format(str(e)))
```

5 项目总结

近年来，自然语言处理的研究已经成为热点，而机器翻译作为自然语言研究领域的一个重要分支，同时也是人工智能领域的一个课题，同样为大家所关注。所以说，我们本次的机器翻译项目是通过机器翻译技术和爬虫技术对“王心凌 B 站视频弹幕进行情感分析”，爬虫是主要技术，基于机器翻译的自然语言处理是关键技术，包括数据预处理、数据库分析、停用词判断、jieba 分词和导入情感分析库等。

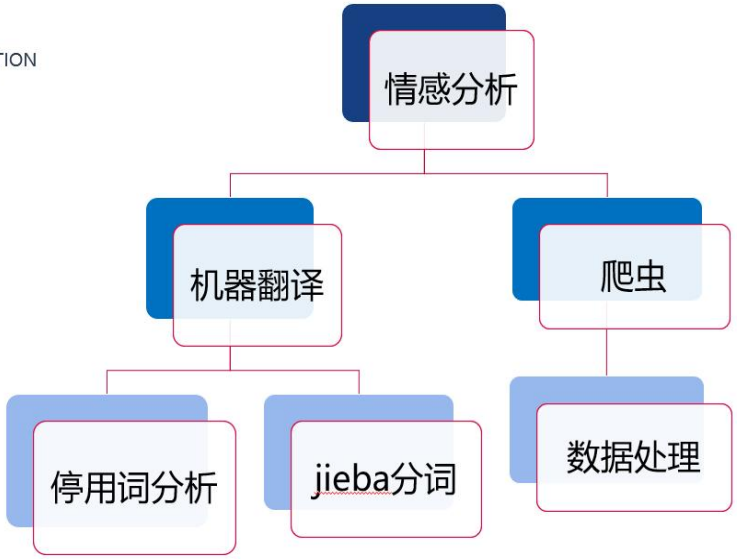

图 5.1 弹幕情感分析流程图

本次项目是基于 python 的，通过 B 站爬虫对王心凌的视频弹幕所做的情感分析，通过导入正则表达式提取文本，爬虫发送请求，获取解析页面后存入 csv 文件，更加具体的步骤是爬取视频地址和对应的 cid 号，进行转换，并且通过数据库分析，导入中文情感分析库，进行词云图绘制并打印出来，再则是通过对大多数评论进行列表排列，通过计数器对它们弹幕的积极、消极和中性弹幕进行判断打分，汇总后对此次分析进行饼图的绘制，直观且快速得出王心凌弹幕-情感分析占比图，将此图显示并保存后把情感分析结果保存到 excel 文件，并据此绘制词云图，首先输入字符串，对停用词进行判断后读取背景图片，设置背景颜色、图片比例及 jieba 分词后生成词云图。

通过机器翻译对身边的事物进行情感分析，可以快速、便利地分析和发现其中的优点与不足，当然，在智能制造时代下，与人工翻译相比，机器翻译是依据基于大量平行语料分析所构建的统计翻译模型进行翻译的，在节省人力、物力、财力资源的同时，它是便捷的，客观的，直接的，它将会应用于更多的研究领域。

参考文献

[1]段炼.面向弹幕文本的情感分析研究[D].重庆邮电大学，2019.DOI:10.27675/d.cnki.gcydx.2019.000121.

[2]姚宗豪.面向B站弹幕情感分析系统的设计和实现[D].山东大学,2021.DOI:10.27272/d.cnki.gshdu.2021.006956.

[3]郭丽蓉.基于Python的网络爬虫程序设计[J].电子技术与软件工程,2017(23):248-249.

[4]李琳.基于Python的网络爬虫系统的设计与实现[J].信息通信，2017(09):26-27.

[5]郑飏飏,徐健,肖卓.情感分析及可视化方法在网络视频弹幕数据分析中的应用[J].现代图书情报技术,2015(11):82-90.