基于节点中心性和机器学习的《复仇者联盟》社交网络研究

宋祎男（2016210485） 王博（2016210522）

北京邮电大学信息与通信工程学院

摘要: 在当今互联网时代，对于社交网络的分析不仅仅局限于单纯的利用节点中心性分析节点的重要程度，用户的情感分析也是当下热门的分析领域。本文基于从猫眼电影网站《复仇者联盟 4》（下文简称《复联 4》）下的爬取 10 万条影评数据，构建了复仇者联盟（下文简称复联）的社交网络，对此网络进行了节点中心性的分析以研究复联中角色的重要程度。并构建了基于朴素贝叶斯算法的情感分析网络，进而得到对观众对于《复联 4》中的角色的情感倾向。通过实验表明，本文提出的朴素贝叶斯算法在情感分析方面有较好的表现。

关 键 词: 社交网络分析; 节点中心性; 情感分析; 朴素贝叶斯模型

A Research on the Avengers' Social Network Based on Node Centrality and Machine Learning

Song Yinan WANG Bo

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications

Abstract: The present society is the era of Internet. The analysis of social network is not only about the importance of node which is analyzed by using Centrality, but also the user's sentiment. Based on the 100,000 pieces of film review of "Avengers: Endgame" from the website of MaoYan Movie, this paper builds a social network of the Avengers. A centrality analysis of nodes is used to study the importance of roles in the network. And the sentiment analysis network based on Naive Bayesian Model is constructed, and then the emotional tendency of the audience to the characters in "Avengers: Endgame" is obtained. The experiment shows that the Naive Bayesian algorithm proposed in this paper has a good performance in sentiment analysis.

Key words: Social network analysis; node centrality; emotion analysis; Naive Bayesian model

在当今时代，随着科学技术日益发达，人们的生活娱乐方式也有了很大的变化。电影已经成为人们生活中最为常见的娱乐方式之一，若是提到当下最火爆的电影，非《复联 4》莫属。截至目前电影已在中国取得了 40 亿的票房，在全球范围内更是成功夺得世界影史票房亚军的宝座。可以想见《复联 4》具有的粉丝基数之大。《复联 4》电影中角色众多，同时每个观众的喜好与情感倾向不同，对电影中角色的情感与态度必然是多种多样。对于此庞大的基于复联角色以及观影粉丝群体的社会网络，粉丝的重点关注角色以及情感分布是一个有趣且值得深入分析的问题。本文便立足于此问题，设计实现了基于中心性度量的社交网络分析、基于朴素贝叶斯的情感分析算法。

基于中心性度量的社交网络分析方法是传统社交网络分析中的常用方法。中心性定义了网络中一个结点的重要性。通过计算度中心性，介数中心性，紧密中心性等参数，综合判断社交网络中节点的重要程度。本文通过分析猫眼电影网站《复联 4》下 10 万条影评，计算上述参数。借此评价复联众多角色的重要性差异。

对于观众对于复联成员的情感分析，采用基于朴素贝叶斯的分析算法，通过机器学习，进行二分类，最终构建情感分布网络，分析出观众对于复联成员的情感倾向。

1 相关工作

本项目主要包含三部分内容，分别是数据爬取，算法实现以及社交网络的搭建与节点中心性的计算。

其中数据爬取以及算法实现部分由王博完成，通过 Python 以及前端解析爬取猫眼电影有关复联 4 的影评数据并做数据预处理，同时搭建基于朴素贝叶斯的机器学习方法获取构建网络所需的相关计算参量。

社交网络的搭建与数据分析部分由宋祎男完成。通过对数据的整理与分析，建立初始社交网络与情感网络，并计算节点中心性以及聚类系数等网络参数对网络进行分析，得到实验结论。

最终的实验报告由两人共同完成。

2 方法

2.1 问题定义

基于本文所研究的节点中心性与情感分析问题，做出以下问题定义：

建立一个由N个节点和M条边构成的无向图，记作 $G=(V,E)$。网络中每一个节点 $v_i \in V(i=1,2,3\cdots N)$，其中每条边表示节点$v$与节点$u$之间有联系。其中对于节点有如下条件定义：

- 用户节点：

  假设猫眼电影网站上爬取用户评论相互独立，无相关性。确保在网络中只有与复联成员节点的连接关系。

- 复联成员节点：

  选取电影中出场率或提及率较高的 18 个复联成员节点作为本文所研究对象进行问题分析。

  所选复联成员之间的关系仅选择电影中重要程度较高的关系进行分析。

  而在进行情感网络分析时，采用朴素贝叶斯模型，假设各特征向量之间相互独立，并假定复联成员之间的情感值均为 1，表示绝对正向情感。

2.2 数据

为了获取关于复联 4 的影评信息，我们选择使用网络爬虫获取相关数据。国内常见的影评信息为豆瓣与猫眼电影，然而由于豆瓣后台服务器的数据限制，每个用户最多只能获取 500 条评论，这并不符合我们复杂网络这一概念，因此我们选择从猫眼电影爬取信息。

首先我们登陆猫眼电影关于复联 4 的主页，寻找影评的有关信息，如图 1 所示：

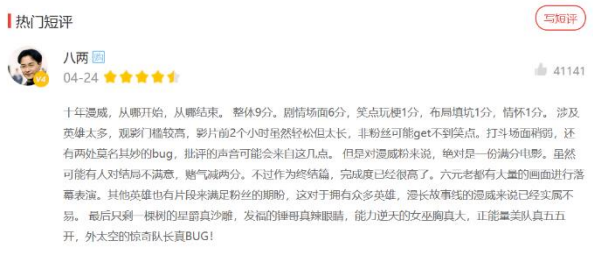

图 1　猫眼电影关于复联 4 的评论内容

不难发现，猫眼电影首页仅为我们提供 10 条热评，这显然不能满足我们的需求。根据爬虫的一般原则，移动端数据往往比 PC 端数据较易获取，因此我们通过登陆移动端的猫眼电影成功进入评论区，其界面如图 2 所示：

图 2　移动端猫眼电影关于复联 4 的评论内容

通过 Chrome 提供的开发者工具，我们发现猫眼电影关于评论的加载是通过动态 json 流加载，其数据含义如下表所示：

数据标签	content	cityName	nickName
数据含义	评论内容	所在城市	观众昵称
数据标签	score		startTime
数据含义	观众评分		评论时间

表 1　数据标签及其含义

由此，我们可以较为方便的构造我们的爬虫程序对其进行爬取。常见的爬虫框架有 pyspider 与 scrapy 等，辅以正则以及 beautifulsoup 等解析工具我们便能较为轻松的获取所需数据，在这里我们仅简要介绍常见的爬虫注意事项，有关爬虫的详细内容可参考文献[1]以及相关资源。

1. 一般情况下，为了避免触发反爬虫机制，我们需要完善请求头信息，其包括的参数如下表所示，我们可以通过 Chrome 提供的开发者工具提取这些参数值：

请求参数	含义
Accept	通告内容类型，表示为 MIME 类型
Accept-Encoding	通告内容编码，压缩算法
Accept-Language	通告客户端理解/优选语言
Connection	控制网络连接是否保持打开状态
Cookie	存储此前发送的 HTTP cookies

Host	请求报头指定的服务器域名和端口
User-Agent	用户代理，确定软件的系列属性

表 2　请求头参数及其含义

其中 User-Agent 一项在爬虫时为必选项，其能有效地避免我们被反爬。

2. 构建代理池

部分网站为了避免被爬虫，设置了较为严格的反爬虫机制，除了限制用户所能浏览的最大信息数目外，其对可疑 IP 也会进行封杀。对此，我们可以选择构建代理池，通过随机选择可用 IP 进行信息的获取，从而避免被反爬，在这里给出一种构建 IP 代理池的 Python 代码：

```
def get_ip_list():
    url = 'http://www.xicidaili.com/nn/'
    headers = {
      'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36'
    }
    web_data = requests.get(url, headers=headers)
    soup = BeautifulSoup(web_data.text, 'lxml')
    ips = soup.find_all('tr')
    ip_list = []
    for i in range(1, len(ips)):
        ip_info = ips[i]
        tds = ip_info.find_all('td')
        ip_list.append(tds[1].text + ':' + tds[2].text)
    return ip_list

def get_random_ip():
    ip_list = get_ip_list()
    proxy_list = []
    for ip in ip_list:
        proxy_list.append(ip)
    proxy_ip = random.choice(proxy_list)
    return proxy_ip
```

通过爬虫程序我们爬取了共 102615 条有效数据，据此我们可构建关于复联 4 主要成员及其评论者的社交网络，并通过计算该网络的相关特征从而获取有关信息，这部分内容将在 3.3 节进行介绍。

2.3 算法

2.3.1 节点中心性

中心性是识别复杂网络中影响力节点的重要的度量工具，通常情况下，中心性指标可对节点是否处于网络中心或者处于网络中心位置的程度进行刻画，进而描述网络中心节点的拓扑结构与特性。本文用到的方法包括度中心性，介数中心性，紧密中心性。下面对这几类方法进行介绍。

本文将构建的网络看作是一个由N个节点和M条边构成的无向图，记作$G=(V,E)$。网络中每一个节点$v_i \in V(i=1,2,3\cdots N)$，其中每条边表示节点$v$与节点$u$之间有联系。而网络结构则用一个$N\times N$的邻接矩阵$A=\{a_{uv}\}$来表示，这里仅考虑无权图，即若节点$v$与节点$u$之间有边，则邻接矩阵对应的值$a_{uv}=1$，否则$a_{uv}=0$。

1. 度中心性

度中心性（Degree Centrality）[2]是对复杂网络中节点互相连接统计特性的最基本描述，其反应了网络演化的重要特征。度中心性是一种简单而有效的节点影响力度量方法，常用于描述在静态网络中节点所产生的直接影响力。一个节点的度中心性等于节点所在网络中与该节点相连的所有节点的数量，也可定义为节点的相邻边数。一个节点的度中心性越高，则该节点与网络中节点的连接强度越大，其影响力越强，在网络中越重要。若用$C_d(v)$来表示节点v的度中心性，则有：

$$C_d(v)=\sum_{u}^{N} a_{uv} \tag{1}$$

其中v表示所要衡量的节点，u为网络中除了节点v以外的其他所有节点。

2. 介数中心性

介数中心性（Betweenness Centrality）[3]主要衡量了网络中节点对于信息流动的影响力，常用于评价节点对于信息扩散的重要性。节点的介数中心性等于通过该节点的最短路径数占所有最短路径数的比例。根据介数中心性的定义，节点的介数中心性越高，则其在信息扩散中负载越重，即经过该节点的最短路径数越多，网络中信息的扩散对该节点的依赖性越高。若用$C_b(v)$来表示节点v的介数中心性，则有：

$$C_b(v)=\sum_{s\neq v\neq t\in V}\frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2}$$

其中σ_{st}为节点s到节点t的最短路径数，$\sigma_{st}(v)$表示节点s到节点t的最短路径中经过节点v的最短路径数。

3. 紧密中心性

紧密中心性（Closeness Centrality）[4]主要刻画了节点到网络中其他节点的难易程度，其等于该节点到其他所有节点距离之和的倒数。一个节点的紧密中心性越高，则该节点到网络中其他节点的距离越小，则

该节点越处于网络的中心，则其影响力越大。若用$C_c(v)$来表示节点v的紧密中心性，则有：

$$C_c(v) = \frac{N-1}{\sum_{u=1}^{N} d_{uv}} \tag{3}$$

其中节点u表示节点集合V中不包括节点v的所有节点，d_{uv}表示节点u与节点v之间的最短距离。当然有的文献中考虑到一般所研究的社交网络中拓扑结构较为复杂，将紧密中心性定义如下：

$$C_c(v) = \frac{N}{\sum_{u=1}^{N} d_{uv}} \tag{4}$$

4. 中心性指标度量对比

中心性指标可通过对节点是否处于网络中心或者处于网络中心位置的程度进行刻画，从而描述网络中心节点的拓扑结构与特性，但其仍存在一定的局限性。

关于度中心性，其只考虑了节点在局部的统计特征而忽略了网络的全局结构，因此可能存在对于节点的影响力衡量不准的问题。若一个用户被微博平台中的许多大 V 所关注，则其在网络中的影响力一定不小，但度中心性则不能很好的区分出这些节点。

关于介数中心性，其度量了全局的所有节点，能较好地解决上文提出的问题，但由于其需要遍历整个拓扑结构从而计算出所有节点对的最小距离，计算量较大。

关于紧密中心性，其同样对拓扑结构全局进行了衡量，但同样计算量较大，且不适用于非全连通图，因为此时存在节点对的距离为无穷的问题。

2.3.2 基于朴素贝叶斯的情感分析模型

文本情感分析（也称为意见挖掘）是指用自然语言处理、文本挖掘以及计算机语言学等方法来识别和提取原素材中的主观信息，常见的方法有文档级情感分类（包括基于监督/半监督/无监督学习的文档情感分类方法），句子级情感分类（包括句子的主客观以及情感倾向性分类）、词语级情感分类（包括基于语义词典和语料库的方法）以及跨语言情感分类等，具体可参考文献[5]。在此我们根据项目需求选择了文档级情感分类，并通过基于监督学习的机器学习方法进行关于复联 4 的影评情感分类问题，该方法由 Pang 等人首先应用于文档情感分类中，具体可参考文献[6].

Scikit-learn(sklearn)是 Python 下的一个通用机器学习库，其主要适用于中小型、实用机器学习项目，通过选择合适的模型我们便可在 CPU 上对模型进行训练。Sklearn 为我们提供了一种模型选择的流程图，如图 3 所示。

由此我们可以根据实际需求选择不同的模型对数据进行处理。对于我们的项目来说，由于我们的数据量为$100\text{K} > [102615 * 5] > 50$，属于文本情感分类问题，且有数据具有标签（观众评分 score），因此我们可以采用朴素贝叶斯模型进行情感分类。

- 朴素贝叶斯模型

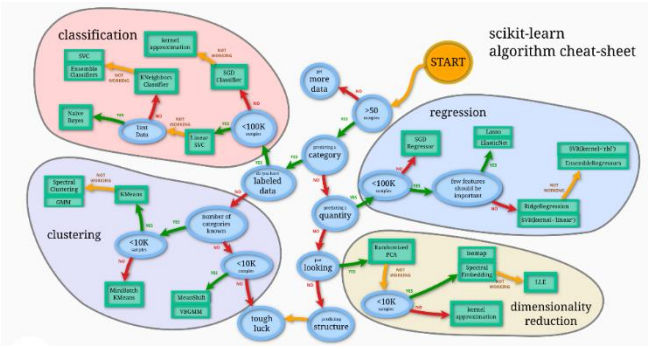

图 3　常见机器学习模型选择法则

在模式分类问题中，利用概率统计中的贝叶斯公式，形成了一套基于贝叶斯框架的决策理论，朴素贝叶斯分类器是贝叶斯分类器中一种简单有效，并在实际应用中十分成功的一种分类器。

朴素贝叶斯模型属于生成式模型（Generative Model），该模型假设文档是由一个参数化概率分布而生成的，建立朴素贝叶斯分类器的两种常用方法包括多元贝努利模型和多项式模型，具体可参考文献[7]。

向量空间模型中，若一篇文档x表示为一个特征向量$(t_1, t_2, \dots, t_N)$，其中N为特征向量的维数，则文档的归类条件概率可以表示为公式(5)：

$$p(x|c_i) = p\big((t_1, t_2, \dots, t_N)|c_i\big) \tag{5}$$

根据朴素贝叶斯分类器的条件独立性假设：在给定类别的条件下，各个特征项之间是相互独立的，则公式(5)可以化简为公式(6)：

$$p(x|c_i) = \prod_{k=1}^{N} p(t_k|c_i) \tag{6}$$

此时，根据贝叶斯公式，分类决策函数如公式(7)所示：

$$c^* = \underset{i=1,\dots C}{argmax}\, p(c_i) \prod_{k=1}^{N} p(t_k|c_i) \tag{7}$$

其中C为所需分类的类别数目，为了防止累乘后计算得到的概率而导致溢出，同时减小计算量，我们常把公式(7)改写为对数形式，如公式(8)所示：

$$c^* = \underset{i=1,\dots,C}{argmax} \left[\log p(c_i) + \sum_{k=1}^{N} \log p(t_k|c_i)\right] \tag{8}$$

在给定了标注样本集后，朴素贝叶斯分类器通常采用最大似然估计（Maximum Likelihood Estimate,MLE）方法来对模型中的参数$p(c_i)$和$p(t_k|c_i)$进行估计:

$$p(c_i) = \frac{M(c_i)}{M} \tag{9}$$

其中$M(c_i)$表示类别为c_i的文档数，M表示文档总数。

对于多项式模型，常采用拉普拉斯平滑（Laplace smoothing）方法进行参数$p(t_k|c_i)$的估计：

$$p(t_k|c_i) = \frac{1 + M(t_k, c_i)}{N + \sum_{k=1}^{N} M(t_k, c_i)} \tag{10}$$

其中$M(t_k,c_i)$表示类别为c_i的文档中所包含特征项t_k的频度。

而对于多元贝努利模型，采用拉普拉斯平滑方法估计参数$p(t_k|c_i)$时如下所示：

$$p(t_k|c_i)=\frac{1+M\left(doc(t_k|c_i)\right)}{2+|D_{c_i}|} \tag{11}$$

其中$M\left(doc(t_k|c_i)\right)$表示$c_i$类文档中出现特征$t_k$的文档数，$|D_{c_i}|$表示$c_i$类文档所包含文档的数目。

由此我们可以给出基于多项式模型的朴素贝叶斯分类器的训练和分类算法，伪代码如下所示：

```
TrainMultinomialNB(C, D)
1   V ← ExtractVocabulary(D)
2   M ← CountDocs(D)
3   for each c ∈ C
4   do M_c ← CountDocsInClass(D, c)
5       prior[c] ← M_c/M
6       text_c ← ConcatenateTextOfDocsInClass(D, c)
7       for each t ∈ V
8       do M(t_k, c_i) ← CountTokensOfTerm(text_c, t)
9       for each t ∈ V
10      do condprob ← (1+M(t_k,c_i)) / (N+Σ_{k=1}^{N} M(t_k,c_i))
11  return V, prior, condprob
ApplyMultinomialNB(C, V, prior, condprob, d)
1   W ← ExtractTokensFromDoc(V, d)
2   for each c ∈ C
3   do score[c] ← log prior[c]
4       for each t ∈ W
5       do score[c]+= log condprob[t][c]
6   return argmax_{c∈C} score[c]
```

图 4　基于多项式模型的 NB 算法的训练及分类过程

其中基于贝努利模型的朴素贝叶斯算法与基于多项式类似，只是部分计算以及参数有所改变，在此不再赘述。

朴素贝叶斯的条件独立性假设声称在给定类别的情况下特征之间相互独立，这对于实际文档中的词项来说几乎不可能成立。此外，多项式模型中还给出了位置独立性假设，而由于贝努利模型中只考虑词项出现或不出现而忽略了所有的位置信息，但即使朴素贝叶斯的概率估计效果较差，其分类决策的效果却出乎意料的好，有关具体论证请参考文献[7]，本文仅应用 NB 算法对影评情感进行二分类。

- 应用朴素贝叶斯分类器进行情感分析

本次抓取的有关复联 4 的影评数据共 102615 条，并选取复联 4 中的 18 位主要成员进行分析。首先我们通过正则表达式+关键词筛选所有影评数据中关于这 18 位成员的评论信息，如图 5 所示。

在获取到不同人物的影评数据后，我们首先需要对自然语言文本进行向量化，这里我们使用 Bag of Words 模型通过构建不同维度数的特征值进行向量化，这里我们不考虑此语和前后词语之间的连接，即每个词都被当作一个独立的特征来看待，这也符合基于多项式模型的朴素贝叶斯思想。

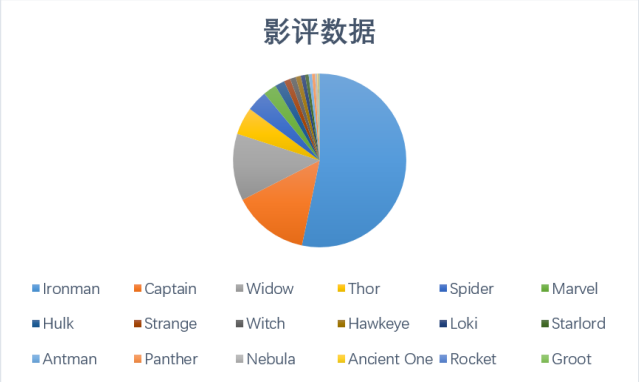

图 5　关于 18 位复联 4 主要成员的影评数据

然而不同于英文、发文等拉丁语系文字，中文天然没有空格作为词语之间的分割符号，因此我们需要使用停用词列表，有关停用词列表请参考文献[8]。

接下来我们正式开始搭建朴素贝叶斯模型，我们首先在网络上获取了常见的有关中文影评的语料库，并通过所爬取数据的 score 标签以及手动标注部分数据形成训练样本，部分数据如下所示：

Comment	Sentiment
I love you 3 thousand，我爱你三千遍	1
托尼.史塔克 有一颗温暖的心[爱心]	1
浪费时间浪费钱 前 2 个小时演的没用的	0
感觉没什么亮点，有惊奇队长还** 差点没打赢。唉，比较失望	0

表 3　训练集部分数据展示

其中 Comment 表示用户的实际影评，而 Sentiment 表示用户评论的情感倾向，其中 1 表示积极态度，而 0 表示消极态度。

通过 Python 提供的 jieba 库我们可以很方便的进行特征向量化，将句子拆分为词语，从而分词后的结果类似于拉丁语系文字，单词之间依靠空格分割，如下所示：

原评论：真的太好看了，尤其是最后全部人联合起来一起打灭霸的时候!!! 看得我超级激动，但是多多少少还是很难过，钢铁侠是我最喜欢的英雄，但是最后却牺牲了，最后打响指说的那句我是钢铁侠，真的泪崩，以凡人之躯，比肩神明
分词后：真的/太/好看/了/，/尤其/是/最后/全部/人/联合/起来/一起/打/灭霸/的/时候/！/！/！/看得/我/超级/激动/，/但是/多多少少/还是/很/难过/，/钢铁侠/是/我/最/喜欢/的/英雄/，/但是/最后/却/牺牲/了/，/最后/打响/指说/的/那句/我/是/钢铁侠/，/真的/泪崩/，/以/凡人/之躯/，/比肩/神明

表 4　特征向量化示例

根据上文所述，在进行初步的分词之后，我们需应用停用词表对其进行处理，从而去除部分对情感分析无帮助的虚词，下面将通过举例来说明停用词的作用：

英文	中文
I love this thing/I hate this thing	我喜欢这个玩意/我讨厌这个玩意
“this” 对情感的分析没有帮助，属停用词	“这次”对情感的分析没有帮助，属停用词

表 5　停用词示例

在去除停用词后，我们仍需对文本进行一定的处理。观察我们的特征矩阵，过于普遍的词汇以及过于特殊的词汇对我们的情感分析没有太大的帮助，类似于某一事物具有极其鲜明的特征但其仅出现一次，这对于我们的情感分析意义不大，因此我们可以继续将特征向量进行降维，至此，我们的评论数据训练集已经完成了特征向量化。

接下来我们应用朴素贝叶斯模型，通过 pipeline 机制连接数据特征向量化与修改参数连接起来，从而通过一次调用降低出现错误的概率，经过一段时间的训练之后，我们首先得到了测试的准确率ρ：

$$\rho \approx 0.8554432 \tag{12}$$

然而对于分类问题，仅仅计算准确率并不全面，而通过混淆矩阵我们能更加清楚的观察到分类效果，其中混淆矩阵如下图所示：

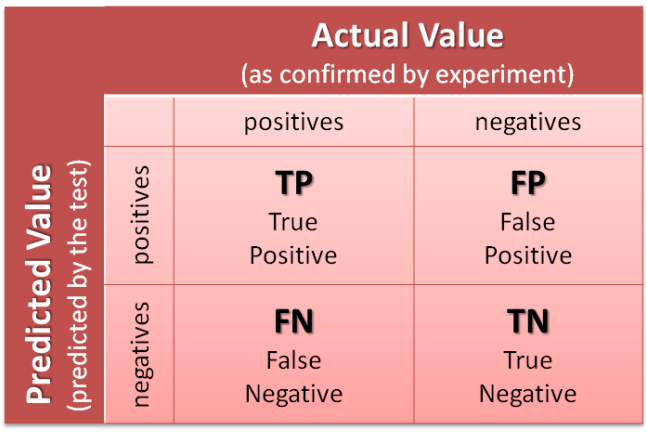

图 6　有关混淆矩阵的说明

其中TP表示本来是积极的，预测也是积极的；FP表示本来是消极的，预测却是积极的；FN表示本来是积极的，预测却是消极的；TN表示本来是消极的，预测也是消极的。根据混淆矩阵我们能明确的看出我们的分类器效果，经计算我们的模型混淆矩阵如下所示：

$$\begin{pmatrix} 3539 & 395 \\ 577 & 2213 \end{pmatrix} \tag{13}$$

从混淆矩阵中我们不难看出我们的分类器对于大部分评论还是能较为准确的对其情感进行划分的，后续我们可以不断增加训练集大小来提高该准确率。

在训练模型后，我们通过 Python 所提供的 joblib 功能对我们的模型进行保存，该功能将我们的模型序列化，通过读取该文件我们便可以轻松对其他数据进行交叉验证了。

后续实际工作中，我们通过输入有关复联 4 的 18 位主要成员的所有影评对其进行分类，同时计算其准确性记录为其$sentiment$值，不难发现$0 < sentiment < 1$，我们可根据该值对复杂网络进行赋权，从而得到有关成员的情感网络，详细工作见 4.1.4 节。

3　实验及结果

3.1　初始复联成员网络度中心性分析

3.1.1　不考虑复联成员内部关系的复杂网络

根据我们所爬取的 102615 条影评，筛选出其中包含不同复联成员姓名的评论构建出一个社交网络，根据实际观影我们构建筛选规则如下例所示：

成员	美国队长
关键词	'美国队长','美队','翘臀','盾','九头蛇','hail hydra','举锤子'

表 6　筛选关键词示例

通过正则表达式我们应用不同的关键词组对影评进行筛选，构建出网络如图 7 所示，该网络反映了复联成员被观众提及的频数，以此作为复联成员节点的连边即产生节点的度。如图 7 所示，复联中最受观众关注的成员为钢铁侠：

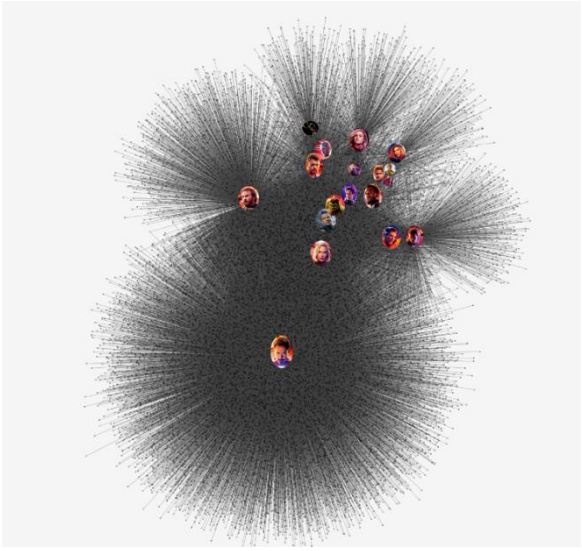

图 7　不考虑内部关系的复联成员网络

通过计算各复联成员的度，如表 7 所示，得到钢铁侠的度为 15067，远高于其余复联成员。其次被观众提及较高的美国队长和黑寡妇的入度分别有 4018 和 3524.但仅仅考虑观众提及次数似乎并不能全面反映一个人物的真实重要地位，因此在下一节我们首先对复联主要成员的内部关系进行分析。

成员姓名	度
Ironman	15067

成员姓名	度
Captain	4018
Widow	3524
Thor	1448
成员姓名	**度**
Spider	1101
Marvel	701
Hulk	514
Strange	338
Witch	300
Hawkeye	269
Loki	226
Starlord	187
Antman	180
Panther	157
Nebula	121
Ancient One	57
Rocket	38
Groot	27

表 7　不考虑内部关系时各复联成员的度

3.1.2　复联成员内部关系网络

通过获取漫威共 3 个阶段 21 部电影的数据，我们可构建这 18 位复联成员的内部网络如图 8 所示：

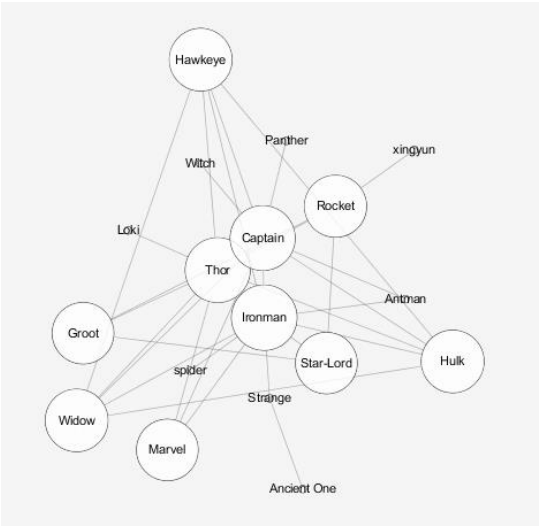

图 8　仅考虑内部关系的复联成员网络

经过生成复联成员内部网络，可以看出，复联中的核心成员为钢铁侠，美国队长和雷神索尔。他们与另外三位黑寡妇，鹰眼，浩克共同组成了初代复仇者，同时这三人能力突出，且有较强的领导力故而处于核心地位。通过对比图 7 与图 8 我们不难发现，身为复联核心的雷神并没有较高的提及率，度仅 1448，与黑寡妇有相反的结果。由此可见，粉丝的提及率更大程度上取决于此次电影中哪个人物更令人感到触动，而非单纯考虑该角色在复联中的地位。

3.1.3　考虑复联成员内部关系的复杂网络

综合粉丝的提及频数和复联的内部成员关系两方面因此，重新构建复联网络，利用节点中心性对其进行分析，其中构建网络如图 9 所示：

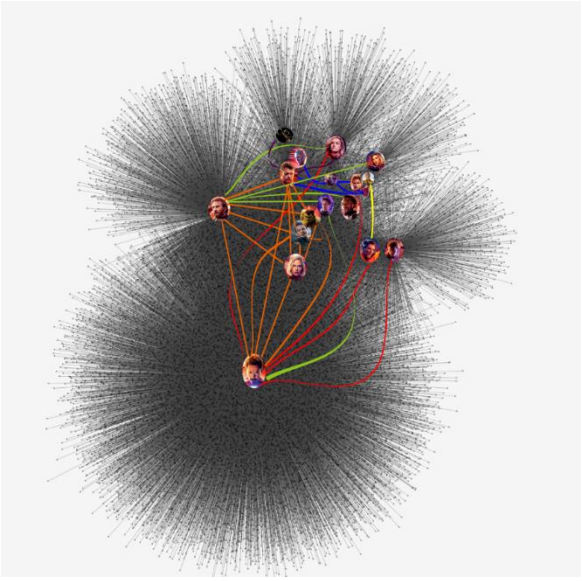

图 9　加入内部关系的复联成员网络

接下来我们对其节点中心性进行分析：

1. 度中心性

根据公式(1)我们对 18 位复联成员的度进行了计算，如表 8 所示：

成员姓名	度
Ironman	15077
Captain	4027
Widow	3529
Thor	1457
Spider	1102
Marvel	704
Hulk	519
Strange	340
Witch	301
Hawkeye	274
Loki	227
Star-Lord	190
Antman	182
Panther	158
Nebula	122
Ancient	58

成员姓名	度
Rocket	42
Groot	30

表 8　考虑内部关系时各复联成员的度

通过直方图我们可以更加直观的观察到各复联成员的重要程度，如图 10 所示：

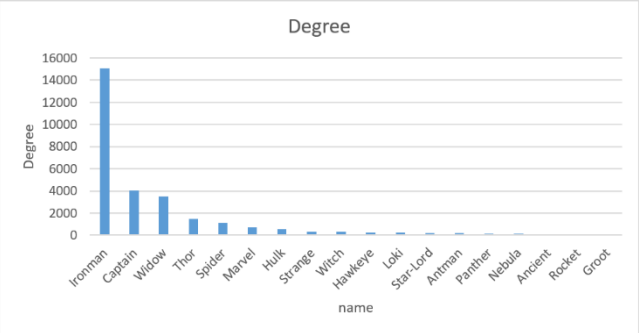

图 10　各复联成员的度中心性

度中心性与节点所连点的个数（在无向图中即度数）直接相关。故而通过测算每个复联成员节点的度数即可反映出该网络中复联成员节点的度中心性。由图 10 可以看出，钢铁侠度最大，也就意味着钢铁侠的度中心性最大，在一定程度上反映了钢铁侠的人气值非常高。同时结合电影情节，钢铁侠在片中的表现也对观众的情感产生较大影响，故而出现在观众评论中的频数较高。

2. 介数中心性

根据公式(2)我们可以计算出各复联成员的介数中心性，如表 9 所示：

成员姓名	介数中心性
Ironman	0.87185947
Captain	0.16265577
Widow	0.09833976
Thor	0.05934998
Spider	0.04549623
Marvel	0.03575437
Witch	0.01249056
Hulk	0.01238566
Strange	0.0111224
Loki	0.00977977
Star-Lord	0.00822323
Hawkeye	0.00411799
Panther	0.00401021
Antman	0.00398419
Nebula	0.00358171
Ancient One	0.00171854
Rocket	0.00081067
Groot	0.00069261

表 9　考虑内部关系时各复联成员的介数中心性

同样我们可以绘制直方图，如图 11 所示：

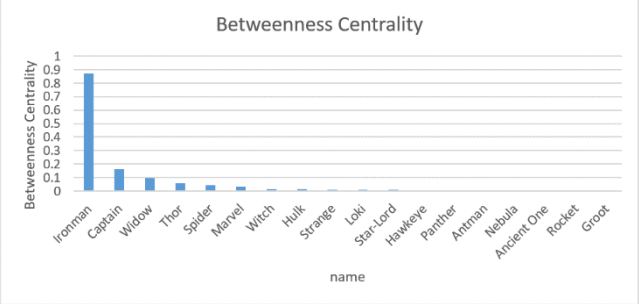

图 11　各复联成员的介数中心性

介数中心性不同于度中心性，它反映的是一个节点担任其他两个节点之间最短路的中间节点的次数。若充当中间节点次数越高，介数中心性越大，相对应该节点的重要性也越高。如图 11 所示，介数中心性的分布与度中心性的分布类似，主要是因为观众在评论的时候，平均提及人数经计算为 1.6 左右，且观众节点之间无联系，完全不相关。因此对于复联成员节点计算其充当中间节点的次数时，若其度较大，即连接粉丝节点较多时，再通过复联成员节点内部连接，可实现其余复联成员与粉丝节点相连，故在此网络中，因钢铁侠度占比较大，度中心性与介数中心性在一定程度上是等价的。

3. 紧密中心性

由于我们的网络规模较大，这里采用公式(4)计算各复联成员的紧密中心性，如表 10 所示：

成员姓名	紧密中心性
Ironman	0.80689942
Captain	0.54647301
Widow	0.53122093
Thor	0.50856604
Hulk	0.49079786
Hawkeye	0.48777669
Marvel	0.48657861
Antman	0.47084789
Spider	0.46736802
Strange	0.45524681
Star-Lord	0.45320634
Witch	0.36157413
Panther	0.35972503
Loki	0.34439083
Rocket	0.34353766
Groot	0.34299141
Ancient One	0.3359928
Nebula	0.33498982

表 10　考虑内部关系时各复联成员的紧密中心性

其数据的直方图如图 12 所示：

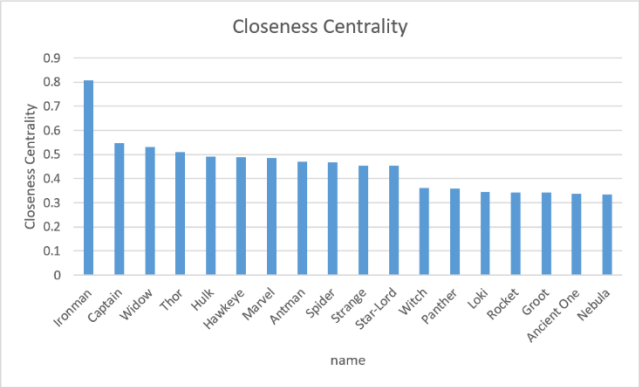

图 12　各复联成员的紧密中心性

对于紧密中心性，求得数据分布如图所示。紧密中心性反映了网络中某一结点与其他节点之间的接近程度。将一个节点到所有其他节点的最短路径距离的累加起来的倒数表示接近性中心性。即对于一个节点，它距离其他节点越近，那么它的紧密中心性越大。因为本实验中的复联网络，钢铁侠的度占比很大，因此，与钢铁侠节点相连的节点，其与其他节点之间的距离就相对来说更近，因而出现与度分布不同的结果。但本身节点度较大的几个复联成员节点，如钢铁侠，美国队长，黑寡妇等受影响较小，紧密中心性仍较高。

综合以上三项中心性，在此社交网络中，钢铁侠，美国队长，黑寡妇等节点三项数据始终较高，说明了这些节点在这个网络中的重要性较高。

4. 聚类系数

聚集系数是表示一个图中节点聚集程度的系数。该网络求其聚类系数为 0.239，说明该网络的节点聚集程度较小。原因是该网络是利用互不相干的用户评论建立的网络，每个用户评论中平均提及对象约为 1.6 个即关联节点少，很少形成三角闭合回路。故而网络聚类系数小。大多数节点处于较为分散且无关的状态。

3.1.4　构建复联成员的情感网络

如 3.3.2 节所述，将爬取的用户影评进行基于朴素贝叶斯算法的情感分析，通过机器学习对每条评论中对于复联角色的情感倾向进行二分类，其准确性作为系数属于(0,1)，由此构建新的情感网络。

首先我们给出各成员的情感倾向比例，如表 11 所示：

成员姓名	积极情感	消极情感
Ironman	0.712266029	0.287733971
Captain	0.818272343	0.181727657
Widow	0.796706417	0.203293583
Thor	0.787983425	0.212016575
Spider	0.950909091	0.049090909
Marvel	0.73962804	0.26037196
Hulk	0.87890625	0.12109375
Strange	0.93452381	0.06547619
Witch	0.882943144	0.117056856
Hawkeye	0.932835821	0.067164179
Loki	0.684444444	0.315555556
Star-Lord	0.762162162	0.237837838
Antman	0.820224719	0.179775281
Panther	0.858064516	0.141935484
Nebula	0.81512605	0.18487395
Ancient One	0.909090909	0.090909091
Rocket	1	0
Groot	0.92	0.08

表 11　影评对各复联成员的情感倾向

为了更加清晰的观察其情感倾向，我们绘制直方图如图 13 所示：

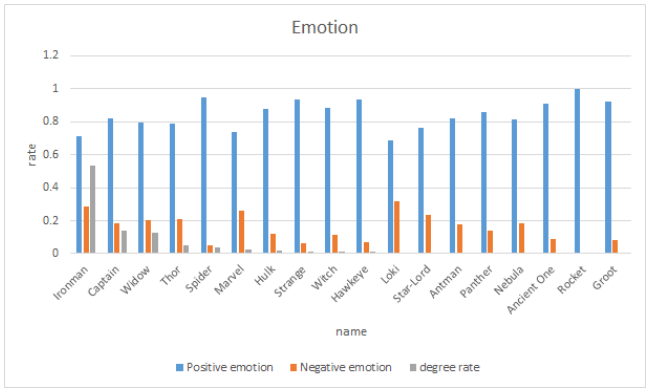

图 13　各复联成员的情感倾向

如图 13 所示，对于复联成员，在观众的整体情感上，明显是更倾向于正向情感，即对复联成员的好的评价居多。对于不同的复联成员，观众的评论提及率是不同的，因此对于如火箭这样的成员，评论数为 37，均为正向情感，大概率是该观众评论是对电影所有角色整体概括，此时情感倾向于正向。而较多的负面情感出现在钢铁侠，雷神，惊奇队长，洛基和星爵均超过了 20%的个人评论数。其中前面 3 人的评论数均超过了 500，而洛基和星爵的评论数为 225 和 185。由此可见，前三者的负向情感更多是真实反映了电影中该角色的行为有值得商榷的地方，因此在观众影评中出现负向情感，而后面洛基和星爵结合电影情节分析，更多的并不是该角色有何错误行为导致观众给出负向情感，而是因机器学习特征数以及样本不足的原因，对于一些伤感情绪如“为什么没有洛基出场”这样的表明为负向，结合语义分析更多的是遗憾的正向情感的判决错误导致。因此对于存在的语义鸿沟问题仍有一些挖掘深入的空间。但从整体来看，基

于朴素贝叶斯算法的情感分析具有较好的结果。能够基本上实现对于语义情感的正确判断，且正确率较高。

现在我们根据情感分析得到结果绘制新的社交网络。这里我们将系数作为网络中边的权值，并将网络中权值属于(0,0.5)的边进行标蓝处理，表示情感倾向于消极情感，将权值属于(0.5,1)的边进行标红处理，表示情感倾向于积极情感。构建得到的网络图如图 14 所示：

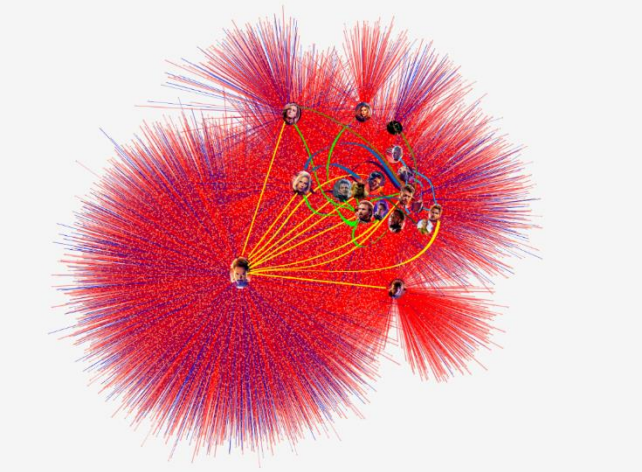

图 14　复联成员的情感网络

我们将在下一节中应用节点中心性方法对该网络进行进一步的分析。

3.1.5　情感网络的节点中心性分析

由于此时的社交网络图已经变成了带权图，部分参数的计算方法已经发生了改变，在此做以简述：

原无权图的邻接矩阵$A=\{a_{uv}\}$需要进行更新，若节点v与节点u之间有边，则邻接矩阵对应的值$a_{uv}=w_{uv}$，其中w_{uv}为该边的权值，否则$a_{uv}=0$。由于邻接矩阵的更新，在计算度中心性、介数中心性与中心性时其取值便会有所改变，下文将给出计算结果。

1.　度中心性

成员姓名	度
Ironman	15076
Captain	4026
Widow	3527
Thor	1456
Spider	1101
Marvel	702
Hulk	517
Strange	338
Witch	300
Hawkeye	273
Loki	225
Star-Lord	188
Antman	180
Panther	156
Nebula	120
Ancient One	56
Rocket	41
Groot	28

表 12　情感网络中复联成员度中心性

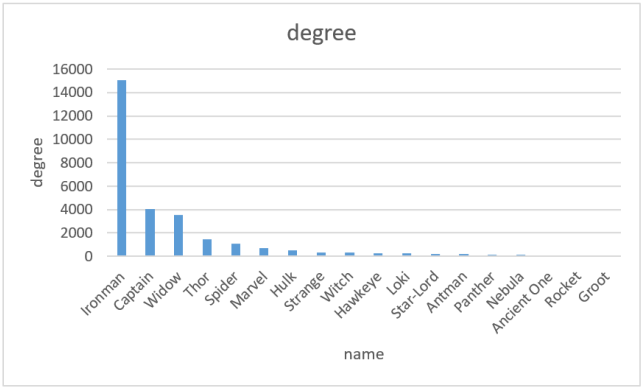

图 15　情感网络中复联成员度中心性

由于我们的权值仍在(0,1)范围内，与原无权图默认权值为 1 相差并不是很大，这导致计算出的度中心性与图 10 相比有所变化但是变化不大。在后续工作中可考虑放大比例从而得到更加明显的对比。

2.　介数中心性

成员姓名	介数中心性
Ironman	0.874817195
Captain	0.163879982
Widow	0.095466187
Thor	0.05790086
Spider	0.045476433
Marvel	0.035871195
Hulk	0.012515
Witch	0.012408157
Strange	0.011027653
Loki	0.009704158
Star-Lord	0.008011974
Hawkeye	0.004182161
Panther	0.004008261
Antman	0.003971603
Nebula	0.003482015
Ancient One	0.001615339
Rocket	0.000811111
Groot	0.000590638

表 13　情感网络中复联成员介数中心性

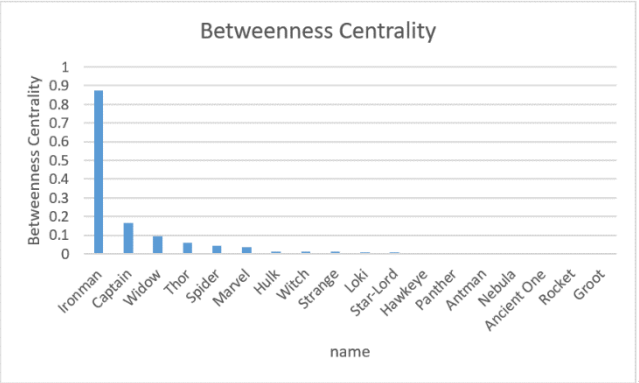

图 16　情感网络中复联成员介数中心性

3.　紧密中心性

成员姓名	紧密中心性
Ironman	0.807314638
Captain	0.546609573
Widow	0.524547175
Thor	0.499433719
Hulk	0.490879652
Hawkeye	0.487868045
Marvel	0.486656634
Antman	0.470837924
Spider	0.469235346
Strange	0.453278262
Star-Lord	0.451482753
Witch	0.359689723
Panther	0.358225032
Rocket	0.342406392
Loki	0.342135622
Groot	0.341592643
Ancient One	0.326927835
Nebula	0.326547682

表 14　情感网络中复联成员紧密中心性

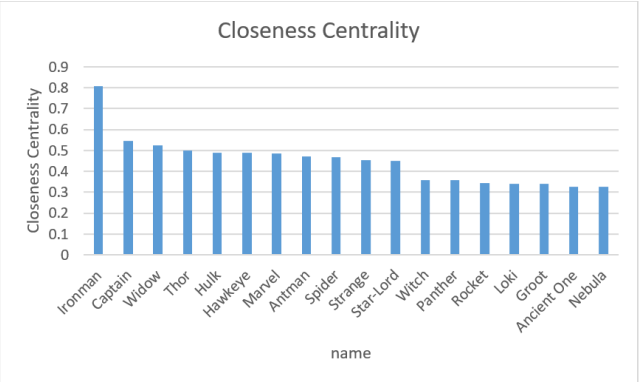

图 17　情感网络中复联成员紧密中心性

结合情感分析可以得到，钢铁侠，美国队长等复联成员的确在观众心中有较高的正义引导性。且复联初代复仇者均有较高的中心性，也说明了复联是以初代复仇者为核心展开的构建。后续加入的复仇者的中心程度相对来说低于初代复仇者。复联在人们心中具有很高的正向地位。由此可以看出，漫威公司在经历了多年的铺垫造就的复联无疑是成功的。因此取得高票房也是自然的事情。观众的评价与其得到的票房有正相关关系。也即《复联 4》是一部无论对于粉丝还是普通观众来说都是一部能够带动情感的值得看的电影。

4　总结

由于本文两位作者均为漫威的忠实粉丝，且动手实现该项目时恰为复联 4 的上映档期，我们便有了构建有关于复联 4 的社交网络，并通过数据分析其他观众对于复联 4 中主要成员的态度。

由于之前有爬虫的经验，因此在爬取数据时还是比较轻松的，通过爬取 102615 条影评数据我们便获取了有效的数据集，接下来便着手于复杂网络的搭建与分析上。

由于是首次接触复杂网络与社交网络的相关内容，我们首先通过查阅资料对复杂网络的基本知识进行学习，考虑到能力与时间有限，我们选择使用 Python 提供对 NetworkX 库搭建网络，并进行一些简单参数的计算，并由此给出一定结论。

考虑到要结合机器学习的相关内容，我们首先通过与老师和同学讨论可能实现的方向，最终决定实现一个文本语义分析中比较热门的领域：情感分析。由于没有可用的 GPU，我们很难对这 102615 条多维数据进行很好的计算，因此最终选择了基于朴素贝叶斯模型的情感分析，最后证明其准确度达到了约 0.856，在有限的训练集的情况下还是可以接受的。

根据我们所搭建的网络以及计算的结果，初代复联 6 人：钢铁侠、美国队长、黑寡妇、雷神索尔、浩克、鹰眼基本是占绝对的重要程度的，而后续漫威电影的主要成员奇异博士、蜘蛛侠、惊奇队长亦有不错的表现，这也与两位作者内心的想法一致。通过本项目，我们初步掌握了一定的数据获取与分析能力，并应用所学知识对自己所感兴趣的话题进行分析，这是十分有趣且有意义的。

由于时间有限，我们的许多内容还没有做的十分完备，在后续工作中我们可以考虑计算更多的网络参数如应用 K-Shell 算法以及 PageRank 等分析节点的重要性，关于复杂网络节点中心性的分析方法可参考[9]。由于计算能力有限，我们仅实现了较为简单的基于朴素贝叶斯的情感分析模型，而基于复杂网络的情感分析是一个十分有趣的问题，若有机会除了实现更为复杂的基于机器学习的情感分析（如 Word2Vec 等），还可以实现基于复杂网络的情感分析，具体可参考文献[10]。

参考文献:

[1] 崔庆才. Python 3 网络爬虫开发实战. 人民邮电出版社.

[2] Degree(graph theory). Wikipedia. https://en.wikipedia.org/wiki/Degree_(graph_theory)

[3] Betweenness centrality. Wikipedia. https://en.wikipedia.org/wiki/Betweenness_centrality

[4] Closeness centrality. Wikipedia. https://en.wikipedia.org/wiki/Closeness_centrality

[5] 张璞. Web 评论文本情感分类方法研究[D]. 重庆大学,2015.

[6] Pang B, Lee L, Vaithyanathan S. Thumbs up? sentiment classification using machine learning techniques[C] //Proceedings of the conference on Empirical methods in natural language processing (EMNLP), 2002.79-86.

[7] (美)Christopher D. Manning 等著,王斌译. 信息检索导论. 人民邮电出版社,2010

[8] https://github.com/chdd/weibo/tree/master/stopwords

[9] 王博. 关于复杂网络与社交网络中节点影响力的研究总结. 北京邮电大学复杂网络与在线社交网络报告

[10] 吉红宇. 基于复杂网络分析的人物关系挖掘[D]. 电子科技大学,2017