**Python基于共现提取《釜山行》人物关系**

**一、课程介绍**

**1. 内容简介**

《釜山行》是一部丧尸灾难片，其人物少、关系简单，非常适合我们学习文本处理。这个项目将介绍共现在关系中的提取，使用python编写代码实现对《釜山行》文本的人物关系提取，最终利用Gephi软件对提取的人物关系绘制人物关系图。

**2. 课程知识点**

本课程项目完成过程中将学习：

* 共现网络的基本原理
* Python代码对《釜山行》中人物关系提取的具体实现
* jieba库的基本使用
* Gephi软件的基本使用

**3. 课程来源**

课程使用的操作系统为 Ubuntu 14.04。你可以在我的 [Github](https://github.com/Forec/text-cooccurrence) 上找到针对《釜山行》人物关系提取的全部代码。你也可以直接点击查看[共现网络简单的英文介绍](https://forec.github.io/2016/10/03/co-occurrence-structure-capture/)。

**二、实验原理**

实验基于简单共现关系，编写 Python 代码从纯文本中提取出人物关系网络，并用Gephi 将生成的网络可视化。下面介绍共现网络的基本原理。你可以在我的博客查看对[共现网络简单的英文介绍](https://forec.github.io/2016/10/03/co-occurrence-structure-capture/)。

实体间的共现是一种基于统计的信息提取。关系紧密的人物往往会在文本中多段内同时出现，可以通过识别文本中已确定的实体（人名），计算不同实体共同出现的次数和比率。当比率大于某一阈值，我们认为两个实体间存在某种联系。这种联系可以具体细化，但提取过程也更加复杂。因此在此课程只介绍最基础的共现网络。

**三、开发准备**

打开Xfce终端，进入 Code 目录，创建 work 文件夹, 将其作为课程的工作目录。下载并安装 gephi 。

$ mkdir work && cd work

$ mkdir gephi && cd gephi

$ wget http://labfile.oss.aliyuncs.com/courses/677/gephi-0.9.1-linux.tar.gz #下载

$ tar -zxvf gephi-0.9.1-linux.tar.gz #解压

下载《釜山行》的中文剧本。

$ wget http://labfile.oss.aliyuncs.com/courses/677/busan.txt

安装jieba中文分词。

$ sudo pip2 install jieba

**四、实验步骤**

你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

$ wget http://labfile.oss.aliyuncs.com/courses/677/busan.py

**1.观察文本结构、准备词典**

《釜山行》剧本非常适合文本处理，语言简洁，大致每一段对应一个关键情节。人物较少且易于识别，所以非常适合文本处理的学习，因此选用了《釜山行》作为课程的样例。

由于《釜山行》人物少、关系简单，所以我们可以通过词典指定人物名称的方式做实体识别。你也可以不建立字典并尝试使用某种分词算法或包装好的分词库（如教程使用的jieba），但离开特定词典的针对特定文本的分词效果可能会有很大程度削弱。因此对简单网络而言，建立字典是效率较高的做法。

可以通过各类百科获取《釜山行》的主要人物，你可以在[百度百科](http://baike.baidu.com/item/%E9%87%9C%E5%B1%B1%E8%A1%8C#3)中找到他们的介绍，并将人名写入一个字典中。项目将主要人物的名称保存在文件dict.txt中，你可以通过下面的命令下载这个字典，也可以自己新建一个文件保存。字典dict.txt需放在文件夹work下。

$ wget http://labfile.oss.aliyuncs.com/courses/677/dict.txt

**2.确定需要的变量**

在work文件下创建代码文件busan.py,开始进行python代码的编写。

在代码中，我使用字典类型names保存人物，该字典的键为人物名称，值为该人物在全文中出现的次数。我使用字典类型relationships保存人物关系的有向边，该字典的键为有向边的起点，值为一个字典edge，edge的键是有向边的终点，值是有向边的权值，代表两个人物之间联系的紧密程度。lineNames是一个缓存变量，保存对每一段分词得到当前段中出现的人物名称，lineName[i]是一个列表，列表中存储第i段中出现过的人物。

# -\*- coding: utf-8 -\*-

import os, sys

import jieba, codecs, math

import jieba.posseg as pseg

names = {} # 姓名字典

relationships = {} # 关系字典

lineNames = [] # 每段内人物关系

**3.文本中实体识别**

在具体实现过程中，读入《釜山行》剧本的每一行，对其做分词（判断该词的词性是不是“人名”[词性编码：nr]，如果该词的词性不为nr，则认为该词不是人名），提取该行（段）中出现的人物集，存入lineName中。之后对出现的人物，更新他们在names中的出现次数。

jieba.load\_userdict("dict.txt") # 加载字典

with codecs.open("busan.txt", "r", "utf8") as f:

for line in f.readlines():

poss = pseg.cut(line) # 分词并返回该词词性

lineNames.append([]) # 为新读入的一段添加人物名称列表

for w in poss:

if w.flag != "nr" or len(w.word) < 2:

continue # 当分词长度小于2或该词词性不为nr时认为该词不为人名

lineNames[-1].append(w.word) # 为当前段的环境增加一个人物

if names.get(w.word) is None:

names[w.word] = 0

relationships[w.word] = {}

names[w.word] += 1 # 该人物出现次数加 1

你可以在 with 代码块之后添加以下代码输出生成的 names 来观察人物出现的次数：

for name, times in names.items():

print name, times

运行代码

python busan.py

在实验楼中的显示结果如下图：



**4.根据识别结果构建网络**

对于 lineNames 中每一行，我们为该行中出现的所有人物两两相连。如果两个人物之间尚未有边建立，则将新建的边权值设为 1，否则将已存在的边的权值加 1。这种方法将产生很多的冗余边，这些冗余边将在最后处理。

for line in lineNames: # 对于每一段

for name1 in line:

for name2 in line: # 每段中的任意两个人

if name1 == name2:

continue

if relationships[name1].get(name2) is None: # 若两人尚未同时出现则新建项

relationships[name1][name2]= 1

else:

relationships[name1][name2] = relationships[name1][name2]+ 1 # 两人共同出现次数加 1

**5.过滤冗余边并输出结果**

将已经建好的 names 和 relationships 输出到文本，以方便 gephi 可视化处理。输出边的过程中可以过滤可能是冗余的边，这里假设共同出现次数少于 3 次的是冗余边，则在输出时跳过这样的边。输出的节点集合保存为 busan\_node.txt ，边集合保存为 busan\_edge.node 。

with codecs.open("busan\_node.txt", "w", "gbk") as f:

f.write("Id Label Weight\r\n")

for name, times in names.items():

f.write(name + " " + name + " " + str(times) + "\r\n")

with codecs.open("busan\_edge.txt", "w", "gbk") as f:

f.write("Source Target Weight\r\n")

for name, edges in relationships.items():

for v, w in edges.items():

if w > 3:

f.write(name + " " + v + " " + str(w) + "\r\n")

完成所有代码编写后，运行

python busan.py

在文件夹work下将会生成busan\_node.txt和busan\_edge.node。

**6.可视化网络**

前面对《釜山行》剧本中的人物关系数据进行了处理，下面我们将使用gephi这个软件来将人物关系可视化，以便展示的更直观，毕竟生硬的数字和文本，或许只有你才懂，其他人可看不明白。

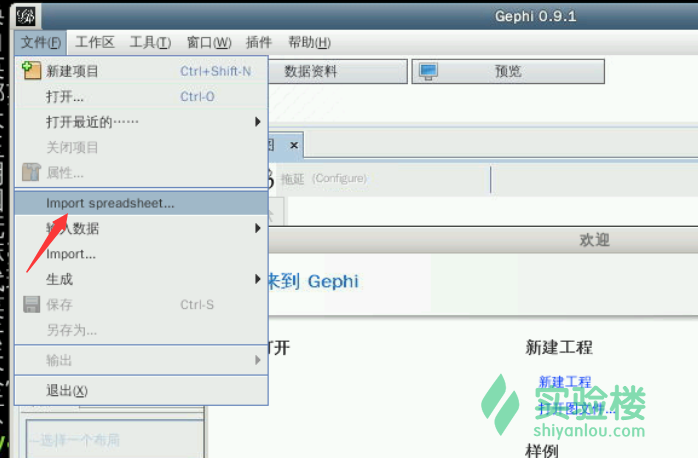
使用 gephi 导入生成的网络，并生成简单的可视化布局。执行下面命令启动 gephi 。

$ cd gephi-0.9.1

$ cd bin

$ ./gephi

点击 文件 -> Import SpreadSheets 。

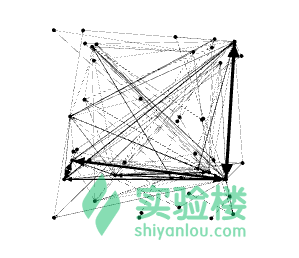


分别选择**节点表格**和**边表格**导入上面代码中生成的两个文件，分隔符选择 空格 ，编码选择 GB2312 。

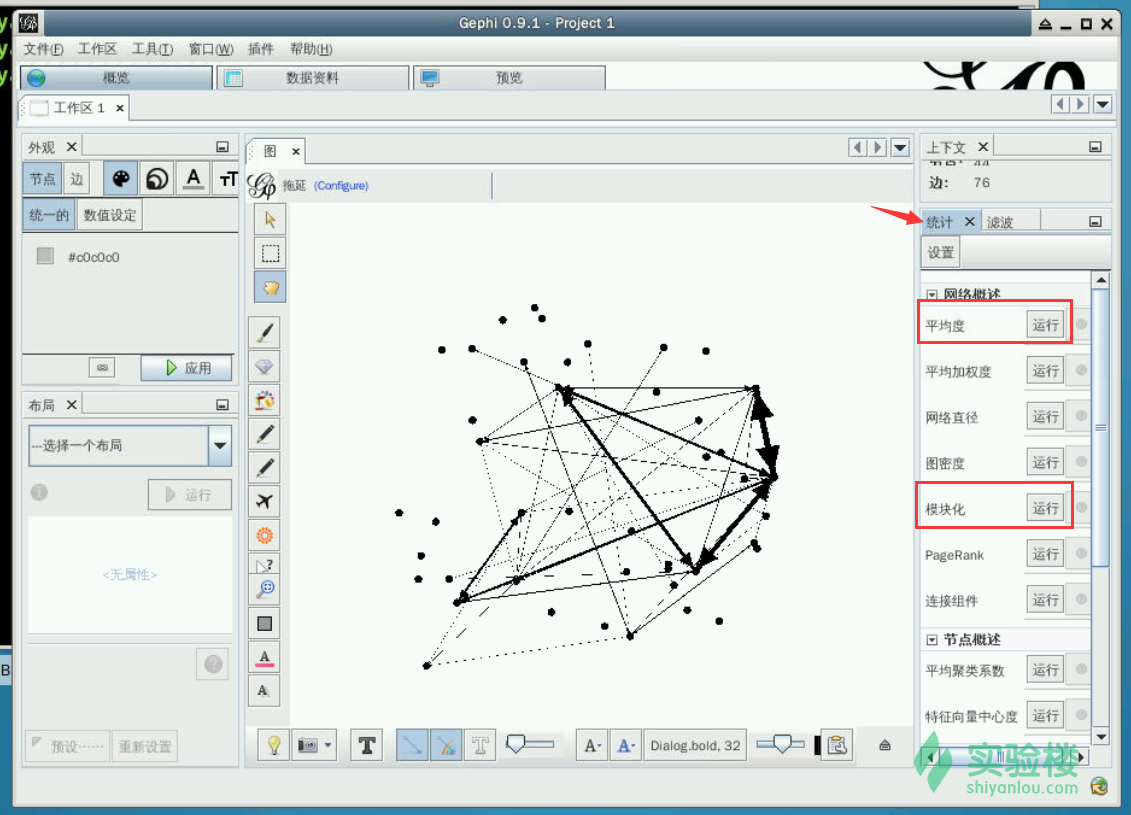




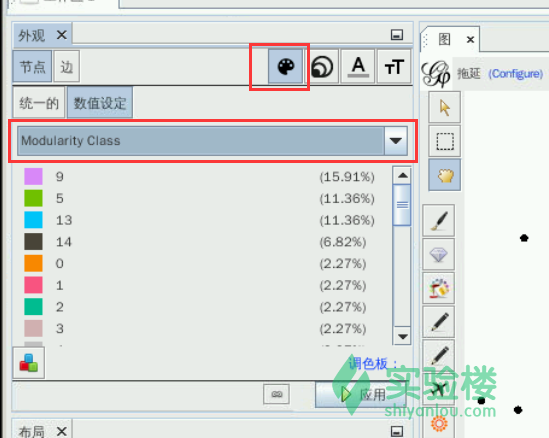
导入后 gephi 将显示所有节点。此时节点没有合适的布局，如下图。你可以在最上方的 数据资料 选项卡中查看图中所有的边和节点，对于分词不准确导致的噪音，可以手动删除。



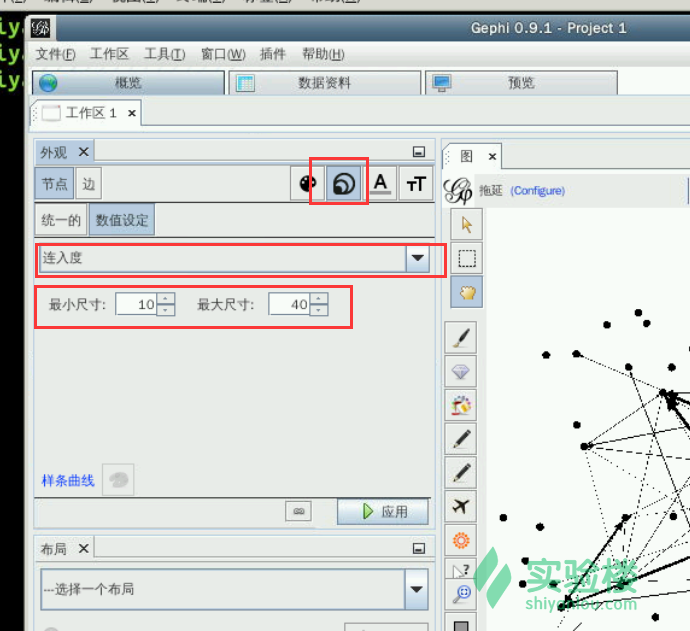
分别点击右侧 统计 栏中 平均度 和 模块化 运行计算。模块化运算时 Resolution 值填写 0.5 。



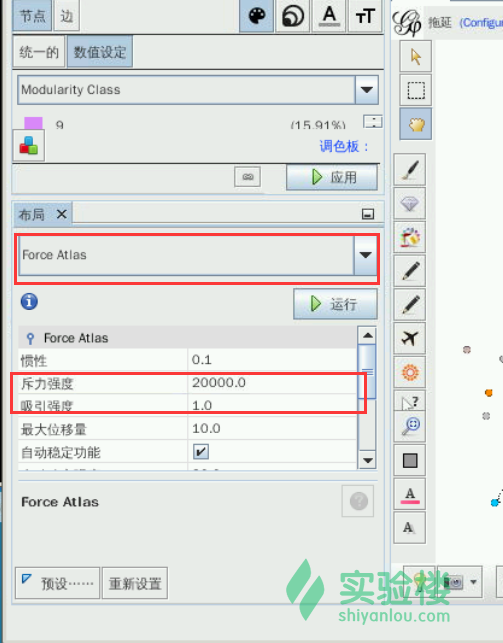
点击左上角外观中节点第一个选项卡，选择数值设定，选择Modularity Class。



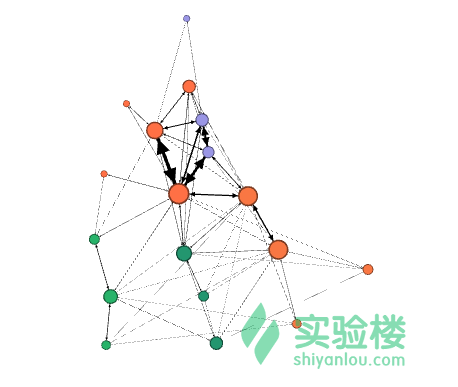
选中第二个选项卡，选择数值设定，选择连入度，最小尺寸填10，最大尺寸填40，点击应用为节点染色、放大。



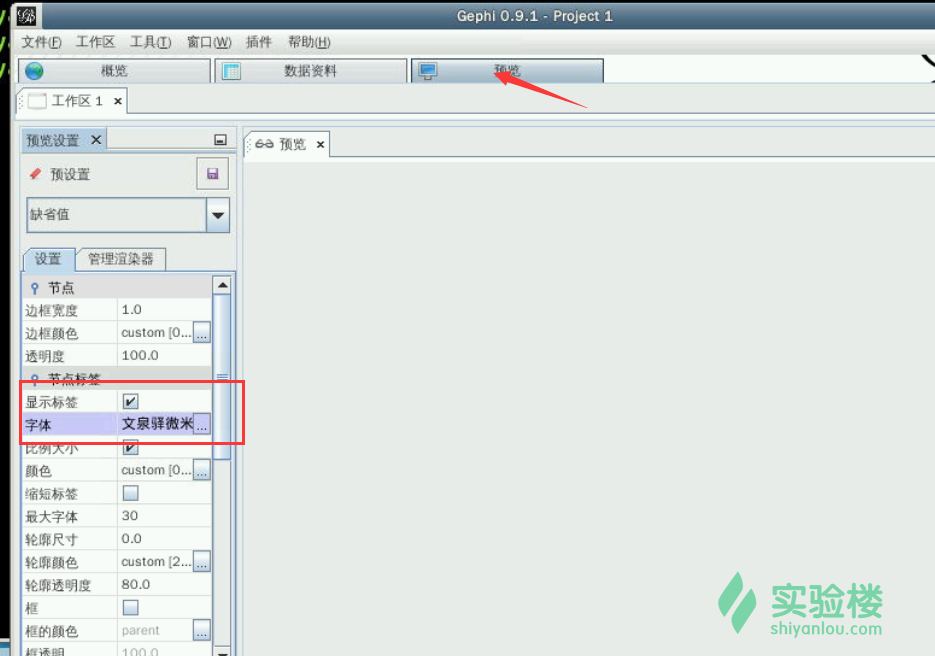
选择左下角布局中的 Force Atlas ，斥力强度填写 20000.0 ，吸引强度填写 1.0 。点击运行，稍后点击停止。



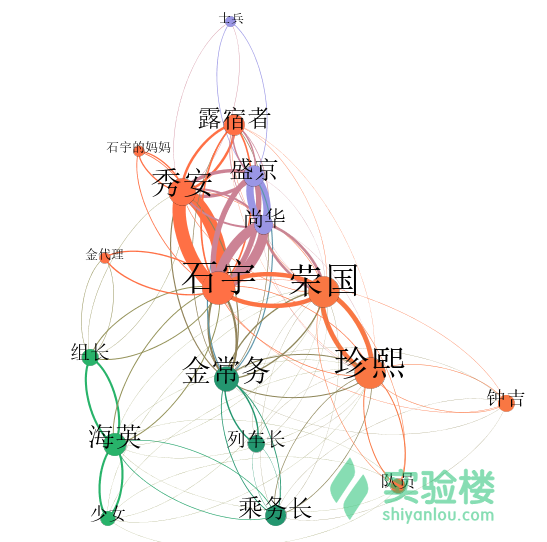
此时布局大致如下图所示。节点染色根据模块化计算结果不定，但染色效果大致相同。



点击最上方的 预览 按钮，选中左侧 节点标签 中 显示标签 选项，并选择一种字体，这里选择 文泉驿微米黑 。



点击刷新按钮，右侧显示最终的人物关系图。



**五、代码获取**

你可以在我的 [Github](https://github.com/Forec/text-cooccurrence) 中获取到完整的代码。如果你有建议或想法，欢迎提 PR 沟通。