University of Hawai'i

# An Empirical Study on the Lord of the Rings Cinematic Universe

*Author*
Selim Karaoglu

*Professor*
Daniel D. Suthers

May 9, 2018

## Abstract

The Lord of the Rings is a high fantasy novel series written by JRR Tolkien. These novels later adapted to cinema with three movies. This project focuses on the cinematic universe, analyzes the structure with scene by scene analysis of the movies, characters with details and some other factors that effect the relationship between characters. These variables are collected and used to create a network to see the structure of the relations in the Lord of the Rings cinematic universe.

# Introduction

The Lord of the Rings novels adapted to cinema with a movie for each book. Later extended versions of the movies presented with cut scenes that are not shown in the movies and some other features. These extended versions of each movie are the published source of the cinematic universe, therefore source of our project too. This project's main objective is to create a network of the extended versions of the Lord of the Rings movies and comment on the results of different mathematical calculations to have a better understanding of the cinematic universe. To achieve this goal, an approach of several steps is set. First, each movie scene is recorded with the characters made appearance in it. Each character titled with their names in the movie are stored. Furthermore some other environmental elements like jewelry, weapons etc. that effect the relation between 2 or more characters are included in the dataset. Second step is to create a network from this dataset. Using Gephi, a network created with all these information in it. After with igraph, using bi-partite projectability of the graph, a charachter-character graph created with weighted edges that represent the relation of the characters. Further several metrics calculated to analyze the structure of the graph. Examining the analysis results is the last step of the project.

# Background

## The Lord of the Rings Cinematic Universe

The Lord of the Rings cinematic universe contains three movies. Each movie share the same name with the books; "The Fellowship of the Ring", "The Two Towers" and "The Return of the King". All three movies later released with extended versions. These extended versions have cut scenes that are not shown in the movies due to the long durations. This project's context is limited with extended versions of these three movies.

## igraph

To calculate the metrics and do some of the plotting, igraph library is used in this project with RStudio. igraph is a collection of network analysis tools with the emphasis on efficiency, portability and ease of use. igraph is open source and free and can be programmed in R, Python and C/C++[7].

## Gephi

Gephi is an open-source software for network visualization and analysis. It helps data analysts to intuitively reveal patterns and trends, highlight outliers and tells stories with their data. It uses a 3D render engine to display large graphs in real-time and to speed up the exploration. Gephi combines built-in functionalities and flexible architecture to explore, analyze, spatialize, filter, cluster, manipulate, export all types of networks. Gephi is based on a visualize and manipulate paradigm which allow any user to discover networks and data properties. Moreover, it is designed to follow the chain of a case study, from data file to nice printable maps[8].

# Methodology

To achieve the goals of this project, the data gathering part is the most important point to get correct results. The methodology of how this data gathered and how the network is created are presented in this section. Since the background knowledge required to understand the methodology is already presented in the previous section, technical details about the process expected to be clear.

## Data

The data collected for the network can be seperated to three main types; scene, character and other.

Scene represents the different scenes from the extended versions of all three movies. There are 169 scenes in our dataset. 37 of them belong to the Fellowship of the Ring, 64 of them are from the Two Towers and the rest 68 of them are gathered from the Return of the King. Each scene record has several attributes; they have unique ids, unique names, character column that contains all characters appeared in that scene, place attribute that shows where that scene took place and date attribute that shows when that scene happened. Date attribute is not available for every scene since dates can't be obtained from the movies. Available dates are obtained from the novels. There are inconsistencies in the movie scene arrangements and novels, therefore obtaining the date of each event is not possible.

Character data of this project contains characters that are credited in the movies with addition of important beings in the movies that are not played or voiced by any actor. To represent them, the names to described them in the novels are used. The creature called Balrog in the movies represented with its name in the novels, Durin's Bane. There are several other characters like Durin's Bane in the dataset. The information about the characters are obtained from both the movies and books. This dataset contains character names, aliases, race, gender, titles (for who have titles), weapon, culture, realm, spouse, birth and death times for each character. Obviously not every character have all of those attributes.

There are several things in the movie that the appearance of it creates a vision or a flashback with other characters, which can't be represented in a same scene due to the fact that the character in the vision is not actually there. But that also effects the relations between characters and should be presented in the network. This is how these things stored in this project's data. And just like scenes, those contribute to the structure of the network.

Finally, the network created with the previous steps is divided into two different networks to analyze. First network contains all data, therefore a multipartite network. Second network is a character-character network. This is created with igraph's biparite_projection() function.

### Graph Structure

To examine the graph structure, several centrality and distance based metrics are calculated. Global and local transitivity, mean distance, degree and weighted degree (in and out, degrees and weighted degrees for bi-partite graph included), eigenvector centrality, pagerank, hubs and authourities, betweenness, closeness, reciprocity, assortativity degree (comparison with rewired graphs), max cliques and community detections (infomap and louvain) are calculated also link communities added with two new graphs.

## Experiments and Results

The methodological process applied and results are examined in this section. With every calculation, an explanation about the process and a comment on what results mean are presented for each process. All calculations are made in the R, using igraph and linkcomm libraries.

## Network

The network of the gathered data is created with Gephi. With the node list of all elements (scenes, characters and other) and the edgelist created by scene relations and other relations, vertices and edges are defined and this multi-partite graph is exported from Gephi. Using the igraph library, first a bi-partite projection created by nodes that symbolize characters and nodes that are not. With igraph's bipartite_projection function, the one-mode network that shows the relation between the characters is saved and used as the second network. This second network provides a better point of view on understanding each character's role in the cinematic universe owing to the fact that with this one-mode graph now it's clear to see the character relations derived from movie scenes and other things such as weaponry or artifacts.

### Graph Characteristics

Previously it is explained that this project focuses on two graphs derived from same network. First graph is a multi-partite graph that contains every information gathered, the second one is the one-mode character graph that focuses on relation between characters.

The multi-partition graph is simple, directed and unweighted, contains 256 vertices and 884 edges. The direction is caused by characters connecting to the scenes and other environment with directed edges. The mentality behind this is, a character is presented with a node and connects to different (scene or other) nodes with a directed edge.

The one-mode character graph is simple, undirected and weighted, contains 64 nodes and 524 edges.

### Edge Weights

Edge weights show how strong the connection between two nodes. The multi-partite graph has 1 for edge weight for all of its edges. This is also logically a must due to the fact that a character can't appear in a same scene twice (or an artifact gifted between two characters can't be gifted again). Those edges on the multi-partite graph are summed and collapsed into weighted edges for the one-mode character graph. It also symbolizes the relationship between the characters played in the same scene.

In the one-mode character graph, there are several strong connections between characters. The strongest connection is between Frodo and Sam. This is reasonable because they are not seperated in movies except few scenes. With the 59 edge weight between them, the connection between them appears as the strongest connected in the network. Relationship between Aragorn, Gimli and Legolas are the following strongest relation. Gimli and Legolas have 54, Aragorn and Legolas have 54 and finally Aragorn and Gimli have 53 edge weight. Those strong relations made this group the second strongest connection in the movie series. The edge connecting Merry and Pippin have 52 edge weight. Those characters took the third place on relationship strength.

## Basic Metrics

### Degree Distribution

Degree is the most basic metric, the number of a vertices adjacent edges. Degree distributions for each graph presented in Figure 1. The multi-partite graph has a heavy tail distribution on the degrees, which is a common observation in real-world network examples. Highest degree nodes represent the characters; Aragorn with 73, Frodo with 69 and Pippin with 66. Degree and weighted degree values are recalculated on the one-mode character graph. With this new calculation, order of the nodes changed. For this graph highest degree nodes represent the characters; Pippin with 47, Aragorn with 45 and Gimli with 44. At this point it's beneficial to remind that the one-mode character graph is weighted, therefore examining the weighted degree values is better to make the "character importance" comparison. Highest weighted degree values belong to the characters; Aragorn with 408, Gimli with 375 and Legolas with 367. Following this trio, Pippin is the fourth with 366, on the other hand this character was the highest degree node.

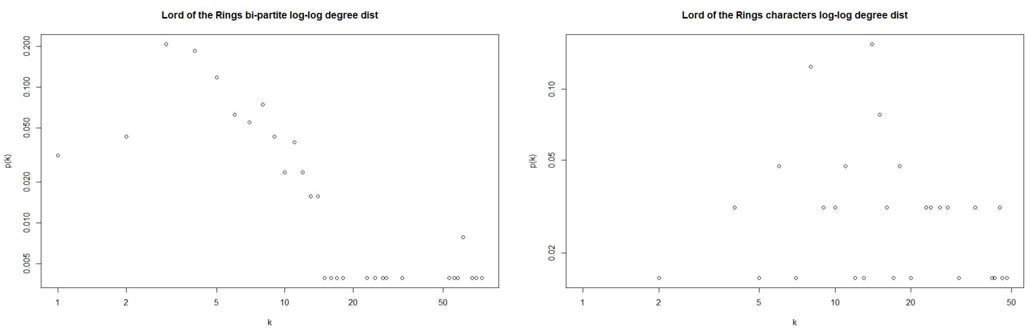



Figure 1: Degree distributions of multi-partite and one-mode character graphs

## Graph Models

There are several graph models created to explain the structure of the graph with random graph creation with some rules. The simplest method is to create a random graph and compare it to this network graph. However, in the previous section the degree distributions are shown, and from the distribution it is clear that this network can't be explained with random graph method. This also implies that a more complicated method needed to explain the degree distribution of this network. The distribution is heavy tailed, therefore power law distributions is a better fit for this concern.

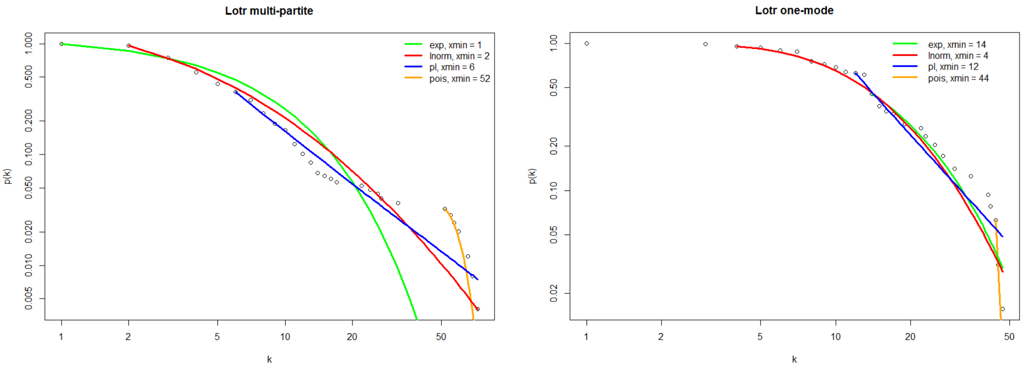



Figure 2: Degree distribution of the graphs with powerlaw distributions

For both multi-partite and one-mode character graphs the closest results belong to log-normal distribution. $k_{min}$ values are also low with 2 for multi-partite and 4 for one-mode character graph. Exponential distribution resulted closer on multi-partite graph. Power law and poisson distributions are not fit to represent the degree distributions on this network graphs.

## Transitivity

Transitivity is also known as the clustering coefficient, is a measure of degree that explains which nodes tend to form a cluster in the graph. Transitivity has two versions, global and local. Multi-partite graph has zero transitivity caused by it's structure.

Local transitivity is the answer to "What fraction of your neighbors are connected?"[1]. In other words transitivity quantifies how close it's neighbors are to being a complete graph. Average local transitivity for the one-mode character graph is 0.7715324. This is a very high transitivity value, and it means that most of the characters in the movie shared at least a scene or other feature with each other. This is reasonable and only exception to that is characters that appeared few times.

Global transitivity is the number of closed triplets over the total number of triplets (both open and closed)[2]. This version was created to give an overall indication of the network's clustering. The global transitivity for the one-mode character graph is 0.5283925. This value is a high transitivity value, and shows that this graph is close to a complete graph.

### Geodesic Distance

Geodesic distance is the shortest path length in a graph. For the one-mode character graph, average geodesic distance is calculated as 1.860119. This means, to reach a node from any other node, shortest path is 1.9 steps in average. That also shows how strongly nodes are connected in the one-mode character graph.

Transitivity and geodesic distance calculations both suggest that the one-mode character graph is strongly connected. This means that the characters shared a scene with lots of other characters in the movies.

## Egocentric Metrics

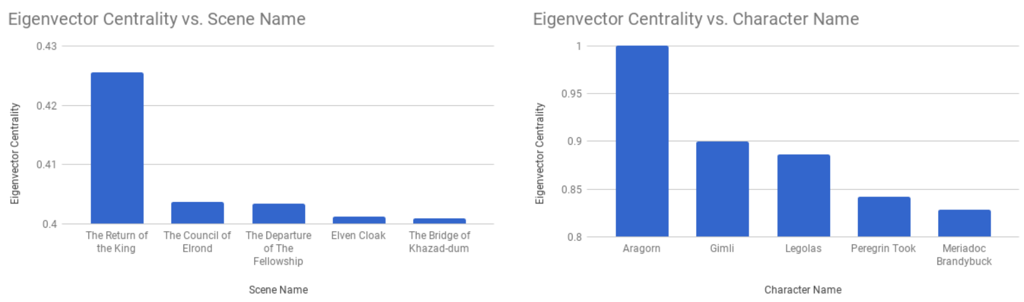



Figure 3: Eigenvector centrality top 5 results of the graphs

### Eigenvector Centrality

Eigenvector centrality is a natural extension to the simple degree metrics[3]. This centrality metric rewards every node for an amount of points for every neighbor it has, but not with the same amount. The amount of points change regarding to the importance of the node. In the previous section, weighted degree was emphasized, now it is possible to compare the results of each calculation. The top scorers for eigenvector centrality are Aragorn with 1, Gimli with 0.95 and Legolas with 0.94. Following the top three, Marry is the fourth with 0.86 and Pippin is the fifth with 0.85. In the weighted degree calculation, Pippin has a higher score than Merry, but eigenvector centrality results differ on that. This implies Merry connected to the nodes with higher importance, therefore has a higher score than Pippin. Figure 3.a shows the top 5 eigenvector values for characters.

The results on the multi-partite graph is the same with the one-mode character graph. However, if the scenes are the point of interest, top three results are; The return of the king with 0.42, the council of Elrond with 0.4 and the departure of followship with 0.4. These scenes are the scenes that most of the important characters were in together like King's crowning or Council in Rivendell. Figure 3.b shows the top 5 eigenvector values of scenes.

### Pagerank

Pagerank gives each vertex some centrality for free, and like eigenvector centrality the amount of centrality points calculated by respecting to the importance of the connected nodes, divided by out-degree of the node. Then vertices that point to many others pass only a small amount of centrality on to each of those others, even if their own centrality is high[3]. With the pagerank calculation, highest degree nodes are; Aragorn with 0.071, Pippin with 0.065, Gimli with 0.064, Merry with 0.064 and Legolas with 0.062. Top 10 pagerank values for one-mode character graph is presented in Figure 4.a. With the out-degrees contributed to the centrality, Pippin became more important comparing to eigenvector centrality results.

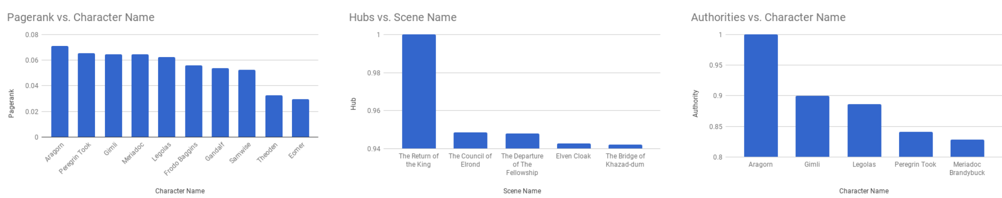



Figure 4: Pagerank, Hubs and Authorities

### Hubs and Authorities

Hubs and Authorities method seperates the importance in two different types. Authorities are nodes that contain useful information on a topic of interest; Hubs are nodes that tell us where the best Authorities are to be found[3]. Applying the same concept to this project's network, authorities are the characters that have high importance and hubs are the scenes that the contains most important characters. This is a natural result of the graph's structure, every edge is directed from character to the scene. Top results for Hubs like presented in Figure 4.b are; The return of the king with 1, the council of Elrond with 0.9485 and the departure of followship with 0.9479. The order is the same with the eigenvector results. Highest Authorities scores belong to; Aragorn with 1, Gimli with 0.899 and Legolas with 0.886, with the same order as the weighted degree and eigenvector results. Top 5 hubs results are shown in Figure 4.c.

### Closeness

Closeness centrality (or closeness) of a node is a measure of centrality in a network, calculated as the sum of the length of the shortest paths between the node and all other nodes in the graph. Thus the more central a node is, the closer it is to all other nodes[4]. To see the closeness of the scenes, multi-partite graph provides the necessary information, and the highest closeness values belong to; the council of Elrond with 0.004132, the eagles are coming and the return of the king with the same result with 0.004115. This is also particularly interesting that those scenes also have the highest out-degree values with the same order. It implies that, those scenes have the greatest variety of characters in them.

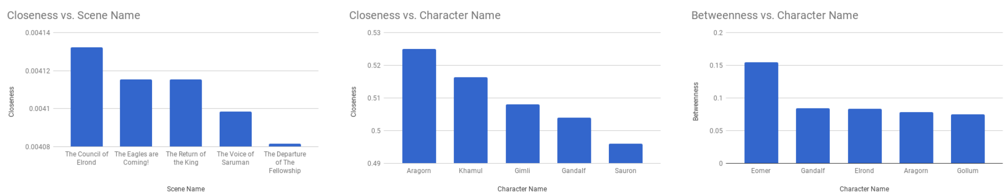



Figure 5: Closeness and Betweenness centrality top 5 results of the graphs

Closeness values for the one-mode characther graph are presented in Figure 5.b. The highest value belong to Aragorn with 0.525. Second highest closeness value belongs to Khamul with 0.516. Khamul is a point of attraction here, because this character is not on top of the list for any other metric.

Khamul's importance on closeness comes from the characters appearance in the movies. This character appears in war scenes that contains important nodes in it. Closeness metric calculations shows the importance of this character even if the number of the characters appearance is not as many as high degree nodes.

#### Betweenness

Betweenness centrality is a measure of centrality based on shortest paths, shows how many geodesic paths crossing that node. High betweenness value means that node is important on connecting other nodes, since number of geodesic paths crossing that node is high. For one-mode character graph, highest betweenness values are; Eomer with 0,154, Gandalf with 0.084 and Elrond with 0.083. Figure 5.c shows the characters with highest betweenness values.

### Structural Metrics

#### Reciprocity

In network science, reciprocity is a measure of the likelihood of vertices in a directed network to be mutually linked[5]. The multi-partite graph has zero reciprocity owing to there is no mutual links in the graph. The one-mode character graph is undirected, therefore reciprocity is 1.

#### Degree Assortativity

Assortativity, or assortative mixing is a preference for a network's nodes to attach to others that are similar in some way. Though the specific measure of similarity may vary, network theorists often examine assortativity in terms of a node's degree[6]. Multi-partite graph's assortativity degree is 0.04. The value is positive and too close to zero, which means there is a correlation between nodes of similar degree, but not too much. For one-mode character graph assortativity degree is -0.23. The negative value indicates there are more similar relationships between nodes of different degree.

Further, the graphs are randomly rewired to see how much of these assortativity values are due to the degree distribution. Rewired multi-partite graph has -0.13 assortativity degree. Rewired one-mode character graph has -0.33 assortativity degree. For each case, assortativity degree values for original graphs and rewired versions are not close. This implies that degree distribution doesn't hold an important position assortativity.

### Community Detection

A network has a community structure, if the nodes of the graph can be grouped easily into group of nodes that each group connected densely. To say more generally, pairs of nodes are less likely to be connected if they do not share communities and more likely to be connected if they are both members of the same community.

There are several community detection methods. In this project, Max Cliques, Infomap and Louvain methods are used. Interestingly, Infomap and Louvain methods resulted exactly the same. The number of clusters for both methods is 4. Figure 6 shows Infomap and Louvain community membership of characters in one-mode character graph. In contrast Max Cliques has 63 number of communities, which is too high for our graph. Modularity is same for Infomap and Louvain with 0.2.

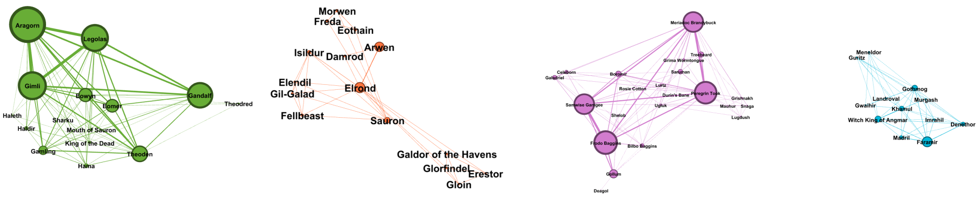



Figure 6: Infomap and Louvain community memberships

To summerize, community detection methods used in this project don't provide a satisfying answer. Max Cliques method is not efficient on community detection, Infomap and Louvain's community detection results can't be presented with any relationship in the movies with an exception of the smallest community can represent the group of characters who took place on Battle of Osgiliath and Minas Tirith.

# Summary

In this project, the Lord of the Rings cinematic universe is studied empirically. First step of the project was to gather the data and create the network. The necessary data is collected from movies by doing a scene by scene analysis and detailed information is obtained with a search from other sources like the novels or notes of the author. The network is designed and two graphs created from the network; a multi-partite graph that shows the character-scene relations, also a one-mode character graph that contains the character-character connections.

The next step is to use the metrics and calculations to analyze the graph and see the features of it. To start with, basic metricss are calculated and degree distribution for both graphs of the network are compared with the power law distributions. In addition to that transitivity and geodesic distance is measured. Egocentric metric calculations are made for eigenvector, pagerank, hubs and authorities, closeness and betweenness metrics. The rankings differ in those metrics, however for weighted degree, eigenvector, pagerank and authorities centralities; the most important character resulted as Aragorn.

Finally structural metrics are calculated. Reciprocity values are not significant due to the structure of the graphs. Degree assortativity is also calculated and compared with random rewired versions of both graphs to see if it's due to the degree distribution. Community detection methods applied to the one-mode character graph to see the community structure lies within. Then the results are compared.

To sum it all, the Lord of the Rings cinematic universe is the point of interest of this project. This universe is analyzed by some calculations and methods. Results are compared and commented to make clearer comments on the graph.

# References


[1] Watts D.J. & Strogatz S.H. *Collective dynamics of 'small-world' networks*, Nature volume 393, pages 440–442. 1998.

[2] Wasserman S., Faust K. *Social Network Analysis: Methods and Applications*, Cambridge: Cambridge University Press. 1994.

[3] Newman M.E.J. *Networks: An Introduction*, Oxford: Oxford University Press. ISBN 0-19-920665-1. 2010.

[4] Bavelas A. *Communication patterns in task-oriented groups*, J. Acoust. Soc. Am, 22(6):725–730. 1950.

[5] Garlaschelli, D. & Loffredo, M.I. *Patterns of Link Reciprocity in Directed Networks*, Physical Review Letters. 93 (26). 2004.

[6] Newman M.E.J. *Mixing patterns in networks*, Phys. Rev. E 67, 026126. 2003.

[7] *igraph Documentation*, Retrieved from http://igraph.org/r/#docs

[8] *About Gephi*, Retrieved from https://gephi.org/about/

# Appendix

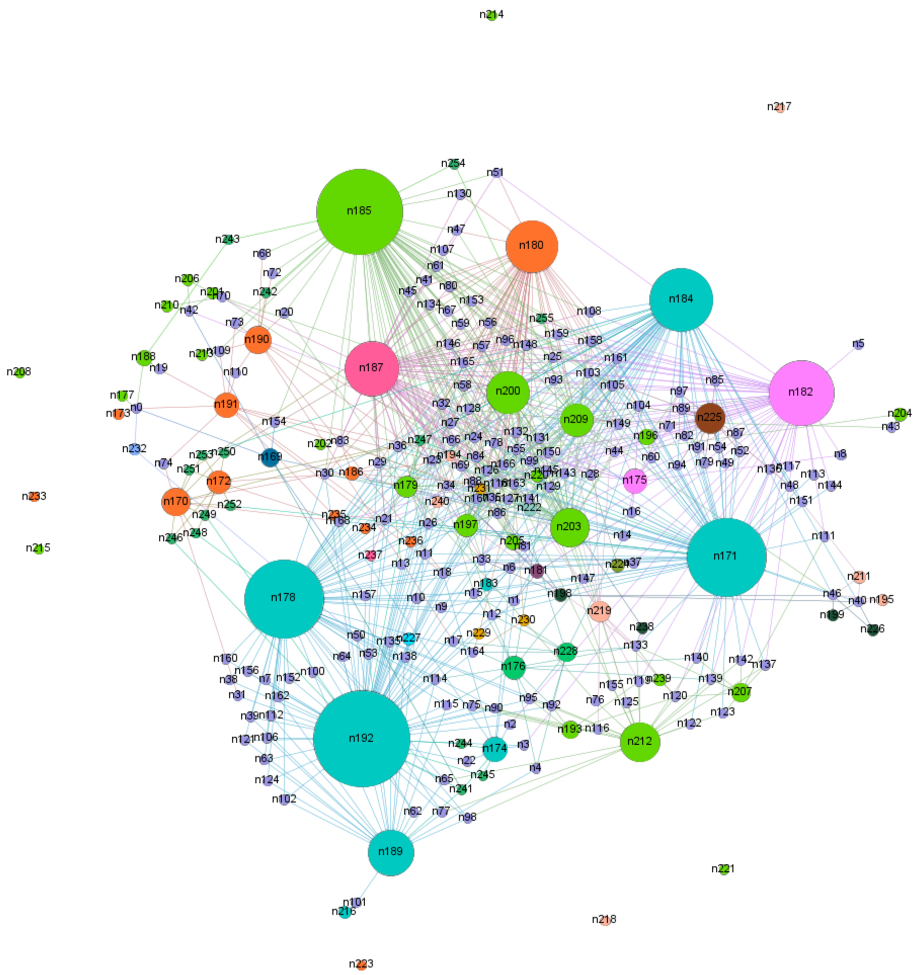
## The R code

```
library(igraph)
library(linkcomm)
library(poweRlaw)
source("PoweRlaw-Utility.R")
source("reify_link_communities.R")
source("new_window.R")
topnv <- function(graph, values, n=10) {
  return(V(graph)[sort.list(values, decreasing=TRUE)[1:n]])
}
############# Graphs
lotr <- read_graph("Lotr-raw.graphml", format="graphml")
V(lotr)$type <- V(lotr)$entry_type == "CHARACTER"
L.bp <- bipartite_projection(lotr)
L.cc <- L.bp$proj2
#Deletion of the characters with no scenes
L.cc <- delete_vertices(L.cc, which(V(L.cc)$Degree == 0))
#Powerlaw and scale free networks
L.cc.degs <- nonzero_degrees(L.cc)
L.cc.disexp <- initialize_disexp(L.cc.degs)
L.cc.dislnorm <- initialize_dislnorm(L.cc.degs)
L.cc.displ <- initialize_displ(L.cc.degs)
L.cc.dispois <- initialize_dispois(L.cc.degs)
plot_discrete_distributions("Lotr one-mode", L.cc.disexp,
        L.cc.dislnorm, L.cc.displ, L.cc.dispois)
lotr.degs <- nonzero_degrees(lotr)
lotr.disexp <- initialize_disexp(lotr.degs)
lotr.dislnorm <- initialize_dislnorm(lotr.degs)
lotr.displ <- initialize_displ(lotr.degs)
lotr.dispois <- initialize_dispois(lotr.degs)
plot_discrete_distributions("Lotr one-mode", lotr.disexp,
        lotr.dislnorm, lotr.displ, lotr.dispois)
#Global and Local transitivity
transitivity(L.cc, type = "global")
transitivity(L.cc, type = "localaverage")
#Mean distance
mean_distance(L.cc)
#Degree
```

```
L.cc.dd <- degree_distribution(L.cc)
plot(L.cc.dd, main="Lord of the Rings characters log-log
        degree dist", log="xy", xlab="k", ylab="p(k)")
lotr.dd <- degree_distribution(lotr)
plot(lotr.dd, main="Lord of the Rings bi-partite log-log
        degree dist", log="xy", xlab="k", ylab="p(k)")
V(L.cc)$i_degree <- degree(L.cc)
V(L.cc)$i_wdegree <- strength(L.cc)
V(L.cc)$eigen_centrality <- eigen_centrality(L.cc)$vector
V(L.cc)$page_rank <- page_rank(L.cc)$vector
V(L.cc)$authority <- authority_score(L.cc)$vector
V(L.cc)$hub <- hub_score(L.cc)$vector
V(L.cc)$betweenness <- betweenness(L.cc, normalized=TRUE)
V(L.cc)$closeness <- closeness(L.cc, normalized=TRUE)
V(lotr)$i_degree <- degree(lotr)
V(lotr)$i_indegree <- degree(lotr, mode="in")
V(lotr)$i_outdegree <- degree(lotr, mode="out")
V(lotr)$i_wdegree <- strength(lotr)
V(lotr)$i_windegree <- strength(lotr, mode="in")
V(lotr)$i_woutdegree <- strength(lotr, mode="out")
V(lotr)$eigen_centrality <- eigen_centrality(lotr)$vector
V(lotr)$page_rank <- page_rank(lotr)$vector
V(lotr)$authority <- authority_score(lotr)$vector
V(lotr)$hub <- hub_score(lotr)$vector
V(lotr)$betweenness <- betweenness(lotr, normalized=TRUE)
V(lotr)$closeness <- closeness(lotr, normalized=TRUE)
reciprocity(L.cc)
reciprocity(lotr)
lotr.rewired <- rewire(lotr, with = keeping_degseq(
        niter = ecount(lotr) * 100))
L.cc.rewired <- rewire(L.cc, with = keeping_degseq(
        niter = ecount(L.cc) * 100))
assortativity_degree(lotr)
assortativity_degree(lotr.rewired)
assortativity_degree(L.cc)
assortativity_degree(L.cc.rewired)
#max cliques
L.cc.mc <- max_cliques(L.cc)
lotr.mc <- max_cliques(as.undirected(lotr,
        mode="collapse"))
for (i in 1:length(L.cc.mc)) {for (v in L.cc.mc[[i]])
```

```
        {V(L.cc)[[v]]$max_clique <- i}}
for (i in 1:length(lotr.mc)) {for (v in lotr.mc[[i]])
        {V(lotr)[[v]]$max_clique <- i}}
L.cc.cl <- cluster_louvain(L.cc)
attributes(L.cc.cl)
modularity(L.cc.cl)
length(L.cc.cl)
V(L.cc)$com_louvain <- membership(L.cc.cl)
L.cc.ci <- cluster_louvain(L.cc)
modularity(L.cc.ci)
length(L.cc.ci)
V(L.cc)$com_infomap <- membership(L.cc.ci)
L.cc.edges <- as_edgelist(L.cc)
L.cc.lc <- getLinkCommunities(L.cc.edges,
        hcmethod = "average", directed =FALSE, plot=TRUE)
lotr.edges <- as_edgelist(lotr)
lotr.lc <- getLinkCommunities(lotr.edges,
        hcmethod = "average", directed =FALSE, plot=TRUE)
write_graph(L.cc.communities, "Lotr-cc-With
        -Link-Communities.graphml", format="graphml")
write_graph(lotr.communities, "Lotr-bp-With
        -Link-Communities.graphml", format="graphml")
write_graph(L.cc, "Lotr-cc.graphml", format="graphml")
write_graph(lotr, "Lotr-bp.graphml", format="graphml")
```

## The Multi-partite graph



## The one-mode character graph