# UNIVERSIDAD POLITÉCNICA DE YUCATÁN


**SOCIAL NETWORK ANALYSIS**

**DIDIER OMAR GAMBOA ANGULO**

**ADRIAN ROBERTO CARMONA RODRIGUEZ**

**PROJECT UNIT 1**

**8A**

**DATA ENGINEERING**

## Abstract

**In this document you will find a deep analysis in the lore of League of Legends, using and interpreting different algorithms used for the analysis of social networks, with the aim of obtaining valuable information in this network**

## Introduction

League of Legends (LoL) is one of the most popular free to play MOBA (Multiplayer Online Battle Arena) games, it was published by Riot Games in 2009, and had in 2016 about 10 million active players. **[1]**

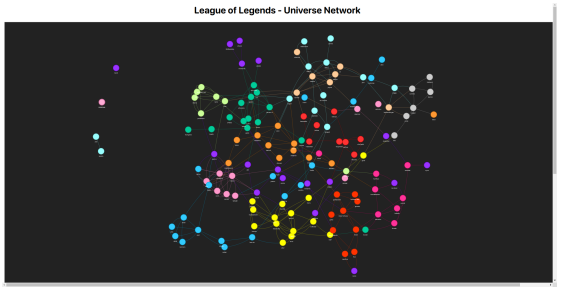
Riot also created the Universe of League of Legends, which contains the history, images, songs, videos, etc. of the planet RuneTerra (where the game takes place), and of the champions and characters that live and interact with it.

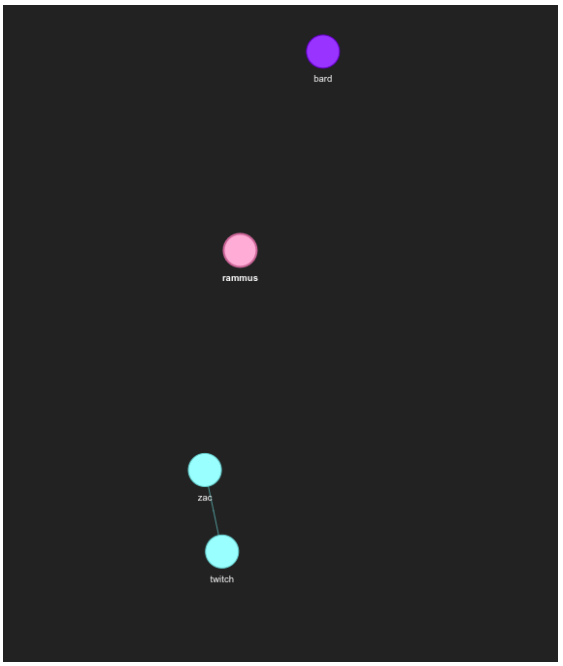
Throughout this document we will develop a social network analysis based on the League of Legends universe, in which we will determine the champions connected by taking as reference the official League of Legends universe in its web page. In addition, information was collected on the Wikia page about each champion to confirm and complete the relationships between them. **[2]**

Officially, the person who developed a first analysis of this network, had access to a riot games API and a complete development by means of R, remaining private the data used in his analysis. So in this paper, we sought to recreate the network completely, from the generation of the data being manually, both regions and champions, to proceed with a deep analysis in Python, with our own development.

## Network characteristics

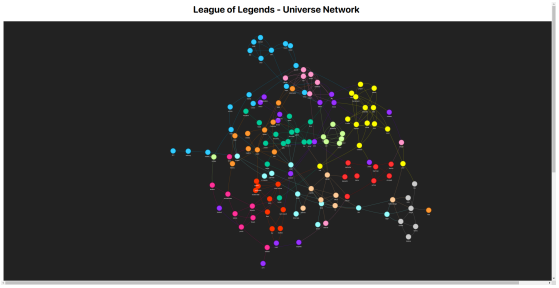
At first glance the original data set has certain nodes which can be considered unrelated to anyone, since if you look at the overview of the Lore, we can realize that there are some champions as "rammus", which do not have any relationship with the other champions, so they are not useful at all and should be eliminated as a first instance.




### Cleaning

Once the nodes have been eliminated, we can say that we have a fully connected network, since each node has at least one connection with which it is related to another node.




## Information & Description

Network size: 305
Number of nodes: 134
Number of edges: 305
Radius: 6
Diameter: 10
Eccentricity based in top 5: [('ahri', 10), ('tristana', 10), ('zyra', 10), ('nami', 10), ('teemo', 10)]
Center: ['warwick', 'camile', 'swain', 'ekko', 'urgot']
Periphery: ['ahri', 'tristana', 'zyra', 'nami', 'teemo', 'lulu', 'kled']
Density: 0.03422735944338458
Average Degree: 4.552238805970149

As an initial study of the network, we can note that it is composed of 305 links and 134 nodes within it, in interpreting the result of the Average degree, we can deduce that on average most nodes have 4 - 5 connections. On the other hand, if we analyze the radius, we can see that 6 is the smallest distance that a vertex can be found from the others. The opposite case is the diameter, which gives us the longest distance between one vertex and another, which is 10 for this network. On the side of the center, we can notice that there are certain nodes which create a subgraph induced by means of the given set of vertices.

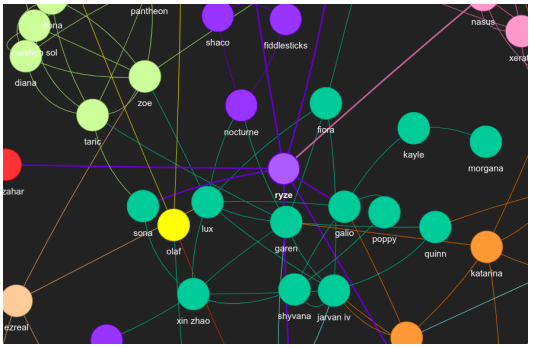
As we can notice, the density of the graph undulates between 0 and 1 according to its classification, if it is a dense graph, its value will be 1 and otherwise it will decrease towards 0. In this case we can notice that our value is 0.03 in principle, so we can deduce that it is not a dense graph.

## Centrality measures

### Closeness Centrality

Top 5 Closeness Centrality
ryze: 0.30296127562642367
swain: 0.29954954954954954
urgot: 0.2982062780269058
nasus: 0.2891304347826087
jarvan iv: 0.2835820895522388

Closeness assigns a node a score based on how close it is to the other nodes in the network. The results that we displayed are the top 5 nodes that are closest in all the network. If you want to choose one, Ryze is the closest node to others.




**Result:**

The principal character to reach all regions in the lore is Ryze. **[3]**


So if I want to read to get a quick idea of the League of Legends universe, I can read the lore of Ryze and those related to him. Because in fact if we look more in detail at the graph, we can see that it is the one that connects us directly with the other regions of League of Legends.

**Betweenness Centrality**

Top 5 Betweenness Centrality
ryze: 0.18391498126640807
swain: 0.16594514427169557
urgot: 0.13470916129197488
miss fortune: 0.11344660938400951
kayn: 0.10442158905991254

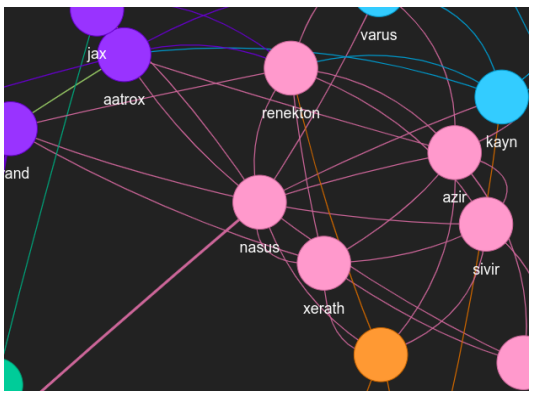
Analyzing and knowing the centrality of betweenness, which is showing us which nodes have the highest flow of information in the network, we can mention that it was logical that ryze appears again, because as mentioned before, it is the node that connects all the regions of lol.

**Result:**

The principal character that has more traffic of information is Ryze again. **[3]**

**Eigen Centrality**

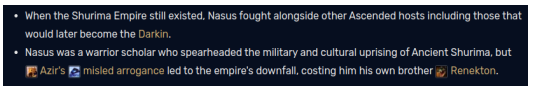
Top 5 Eigen Centrality
nasus: 0.39999221966375337
renekton: 0.3639229697757953
azir: 0.34884104310547903
xerath: 0.2813374132284678
sivir: 0.258058230659677




It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. So in our top 5, nasus takes the top spot and that is because if we analyze the nasus node in the network, we will realize that nasus has many links. But as you can see, most of its links come from the same region, so we can conclude that it is a very important node both in the Shurima region and in the network.

**Result:**

Nasus is the most important node in the Shiruma region. [4]




Well it is true, if we check the lore, Nasus had a lot to do with the uprising and the wars of the Shiruma empire.

**Katz Centrality & Page Rank**

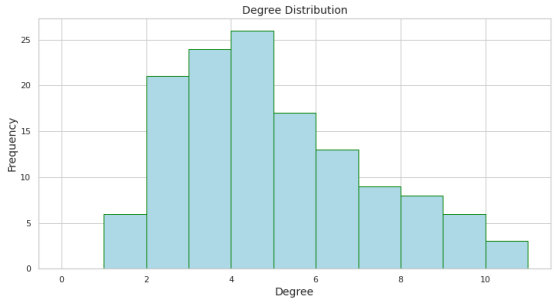
Top 5 Katz Centrality
nasus: 0.199296634349094
renekton: 0.17883984000496286
azir: 0.1686958785518876
aatrox: 0.15202775718035164
kayn: 0.14651437491978614

Top 5 Page Rank
swain: 0.014954840900225043
nasus: 0.014085627953226284
miss fortune: 0.013798487965606109
veigar: 0.013701263972633018
lissandra: 0.013025118750178716

As we can see and as we already know, page rank and Katz are similar algorithms, which try to identify the most influential nodes within a network, and it should be noted that both are variants of the Eigenvector

centrality. Having said this, it is not strange that nasus appears again as the most influential node, even in the page rankings that by thousands of thousands was in second place.

## Degree distribution




In this graph we can notice all the frequencies of the distributions, for obvious reasons we can see that the graph should start at 1, because as we know the degree is calculated by the link count that has a node, then if we know that it is a connected network, it means that at least all nodes must have 1 link, however when we talk about degree distribution, this can be compared with the Average degree, which as we saw earlier was 4. 5 and if we look at the graph, the 4.5 can be said to be the most frequent in the entire network, meaning that most nodes on average have 4 to 5 nodes.

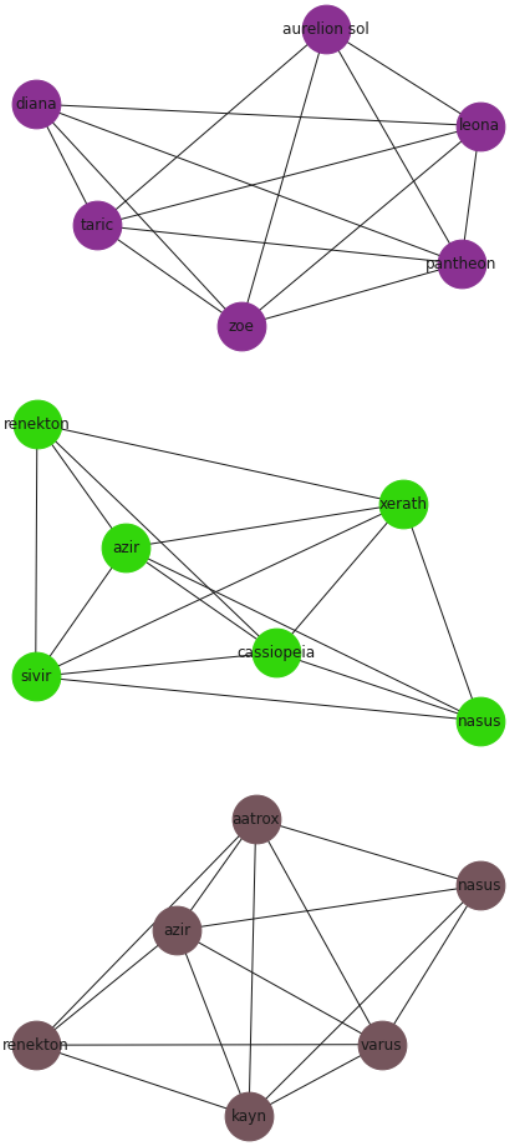
**Result:**

In average and majority nodes have 4 - 5 nodes

## Community detection

### Top 3 cliques in the graph

If we observe the network and we want to search the subnetworks formatted by the strongest connected, then searching the cliques is the best option, for this we find all possible cliques in the network and select only the 3 largest ones.




If you wish to read a lore without going out of context, you can choose any of these groups of characters, so that the story does not expand into other regions or themes in the lore.

### Edge Betweenness

As mentioned above, if we use the betweenness centrality to calculate or identify the communities in our network, it will identify the areas in the network where there is more information flow, that is, the most important areas of the network, and this will be returned to us as communities.

It is worth mentioning that this calculation was made with a library that uses modularity in order to verify that the communities are well connected. Networks with high modularity have strong connections between nodes within modules, but few connections between nodes in different modules.

**Results of community detection with Edge Betweenness :**

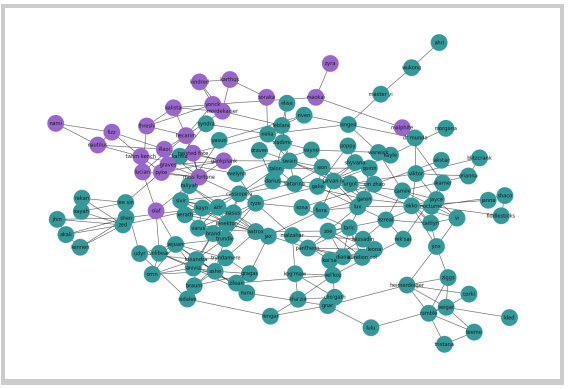
Clustering with 134 elements and 9 clusters
[0] ahri, wukong, master yi, singed, dr mundo, warwick, viktor, camile,
blitzcrank, ekko, orianna, jayce, vi, caitlyn, ezreal, janna, skarner,
rek'sai, malphite, maokai, zyra
[1] riven, yasuo, taliyah, nasus, xerath, azir, sivir, cassiopeia, jax,
renekton, varus
[2] irelia, syndra, karma, lee sin, xayah, rakan, kayn, zed, shen, jhin,
akali, kennen
[3] vayne, leblanc, swain, urgot, sion, evelynn, vladimir, elise, jarvan iv,
draven, darius, katarina, garen, lux, talon, galio, fiora, shyvana, xin
zhao, quinn, nocturne, kayle, poppy, fiddlesticks, shaco, alistar, morgana
[4] soraka, mordekaiser, gankplank, karthus, kalista, hecarim, kindred, tahm
kench, yorick, thresh, illaoi, miss fortune, lucian, graves, pyke, twisted
fate, fizz, nautilus, olaf, nami
[5] jinx, ziggs, heimerdinger, rumble, veigar, corki, gnar, tristana, teemo,
lulu, kled
[6] kassadin, malzahar, kai'sa, vel'koz, kog'maw, cho'gath, zilean, kha'zix
[7] zoe, taric, aatrox, aurelion sol, leona, pantheon, diana, sona
[8] udyr, ryze, sejuani, nidalee, lissandra, brand, trundle, volibear, anivia,
tryndamere, ashe, ornn, rengar, nunu, braum, gragas

**Result:**

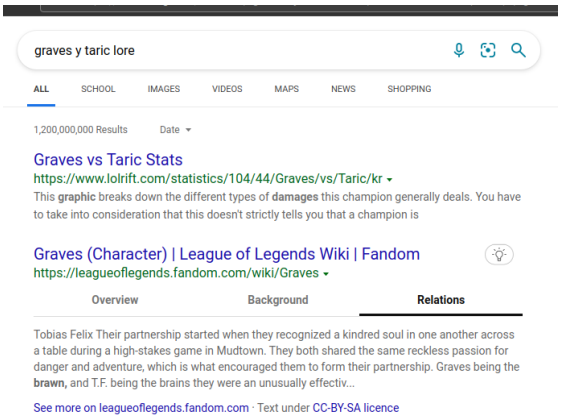
If we want to identify conflicts in the lore of League of Legends, this type of analysis can be the most useful, as it can show us which groups of nodes, i.e. which characters have had fights among themselves without involving others from other regions or from another history.

## Community Detection with Girvan and newman

As we know this algorithm is based on the elimination of edges iteratively according to the largest number of shorter paths between the nodes that pass through them. The result is a total decomposition of the graph, so when dividing it returns smaller groups, which represent the communities.



If we analyze the result of this algorithm makes a lot of sense, the fact is not like the previous one that focused on only the groups that have fought among them, which encompasses something bigger, in addition to taking into account the fighting groups, seeing 2 communities in total, we can notice that if we put it at the level of lords, there would be 2 groups of lords, implying that outside of those communities, the lords no longer relate as much to each other or that at some point history becomes different through the regions, the regions in which each community is involved.




**Result:**

An example of this is to look for Graves that is from one community with Taric that is from another, as we can see in google there is not a single thing that relates them to each other, so their lords will be totally different..

## Conclusión

As we could see in this document there are many things that can be taken into account in an analysis of social networks, but what matters is to understand what you want to find, since there are multiple algorithms to calculate things, but the understanding of what each algorithm gives us back is what adds value to our research, such as which character to pick to tell me a little bit about the League of Legend universe, or which character or group to grab so as not to be playing different stories, who had the most influence on those stories or what are the most relevant fights between groups.

**References:**


1. “League of Legends”, Universe - Champions. Link

2. “Universe of League of Legends Network Analysis” - Camilo Vega Ramírez. 1 Noviembre 2018. Link

3. “League of Legends Wiki” - FANDOM Games Community - “Ryze”. Link

4. “League of Legends Wiki” - FANDOM Games Community - “Nasus”. Link

5. Barabási, Albert-László (2016) Network Science. USA.