Wikispeedia Game Analysis via Network Science

STA404 2021 Network Science and Computing

Wang Jinghui 11813126

Luo Yiling 11812532

Huang Qiyu 11811532

May 30, 2021

Abstract

Wikipedia is a free online encyclopedia that is created and edited by volunteers around the world. It has a great global influence, so it is important to explore the connection between words on Wikipedia. Based on the data of game "Wikispeedia", this article summarizes the characteristics of the Wikipedia network by exploring the Links between words on Wikipedia and proposes one optimal and convenient solution for the path between words on Wikipedia.

Keywords: *wikipedia links* *network analysis* *wikispeedia game*

1 Introduction

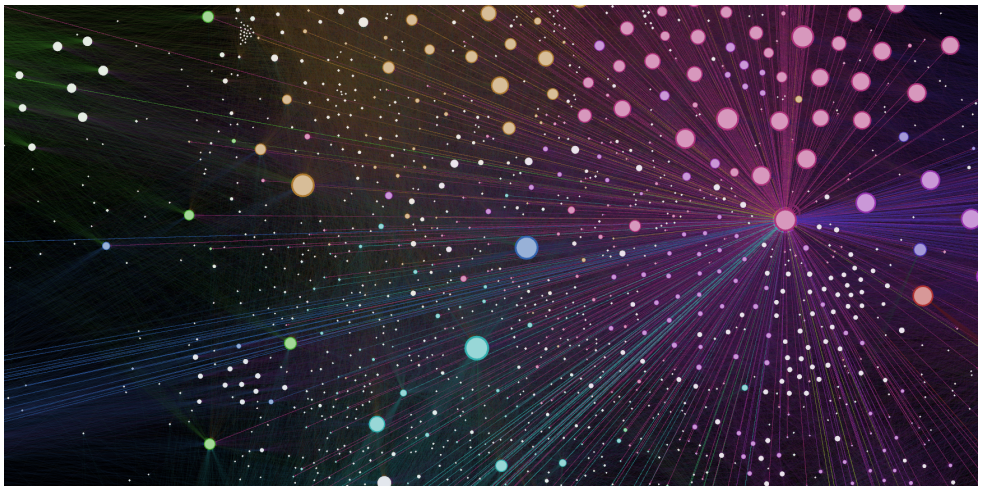


Figure 1: Part of Wikipedia Web Pages Network [3]

Wikipedia is the largest encyclopedia of all time, and it is still growing. Survey data in recent years show that Wikipedia is one of the ten most visited sites on the Web, which has had a great impact on people's lives [2]. The navigation information space provided by Wikipedia is an important part of human daily life.In order to design an efficient and user-friendly navigation information system, it is very important to understand how humans navigate and find the information they need [4]. There is a novel online game called "Wikispeedia", which can be used to explore the network relationship between words on Wikipedia: players have to reach an article from another, unrelated article, only by clicking links in the articles encountered

[5]. Our group hope to explore the Wikipedia network to understand the characteristics of Wikipedia links and to propose efficient approaches to help people to get from one Wikipedia word to another.

2 Basic Statistics

The data 'wikilinks' is a social network that is directed and unweighted, and the direction means the accessible click from the source page to the target page in Wikipedia.

This network has 4592 nodes (represent the definition in Wikipedia) and 119882 links (stand for the connection and direction between two definitions), so the average degree of this network is $\langle k_{in} \rangle = \langle k_{out} \rangle = 26.107 = \langle k \rangle / 2$. The maximum of out degrees is 294 and maximum of in degrees is 1551. In addition, there exist some nodes with zero in degree or zero out degree. The maximum of degrees is 1845.

By computing, we find that this network's diameter d_{max} is 9, which means it takes up to nine steps to get from one point to another. And also the average path length $\langle d \rangle$ for the connected graph is 3.203.

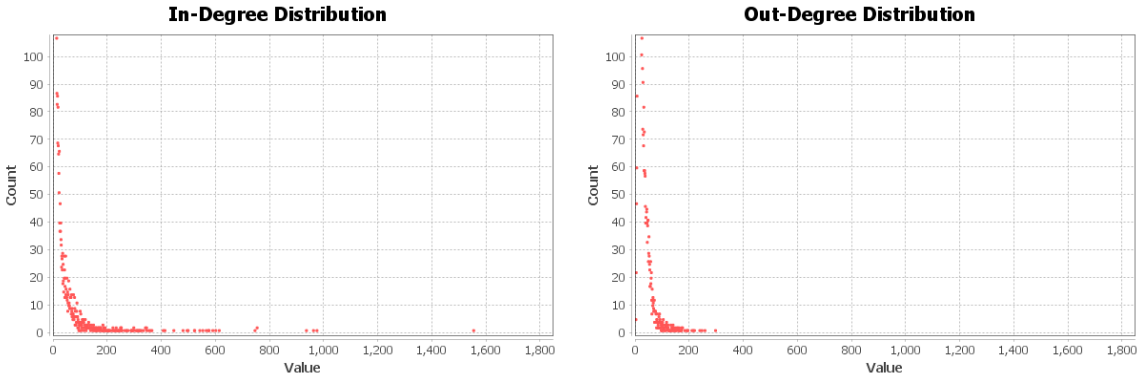
The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together, in this network $\langle C \rangle$ is 0.183. In a real network, hubs turn to have a smaller clustering coefficient. This number is reasonable as some nodes have extremely large degrees, so it's difficult for all their neighbors to have connections.

3 Structure

3.1 Components

We also see that the giant component of this network has 2589 nodes which contain 99.93% of the nodes, so to be simplified and without loss of generality, we delete the isolated points in our later study.

3.2 Degree Distributions



(a) (b)

Figure 2: (a) In degree distribution (b) Out degree distribution

Use log-log axis to plot them, then we get Figures 3.

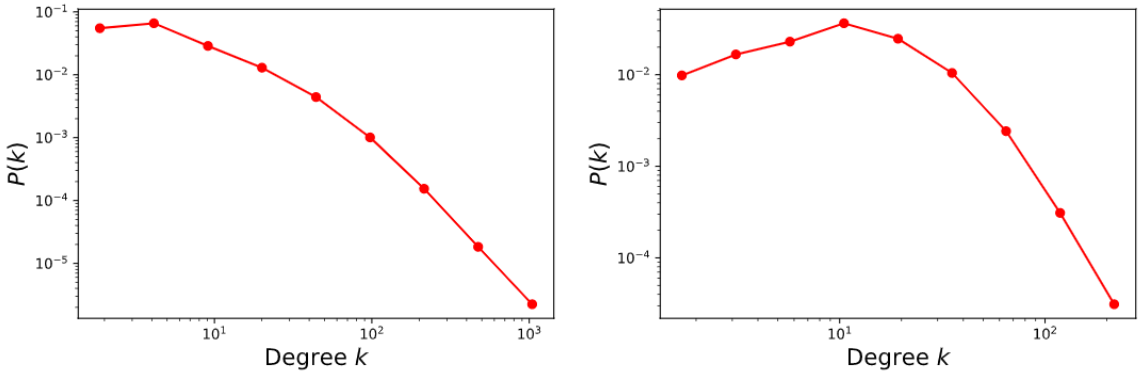
3.3 Connectedness

As $\log N = 8.432$ is smaller than the average degree 26.107, so we denote this network as connected with no cluster size distribution and dense giant component with loops compare to the huge number of nodes and its directed property.

3.4 Scale-free Property

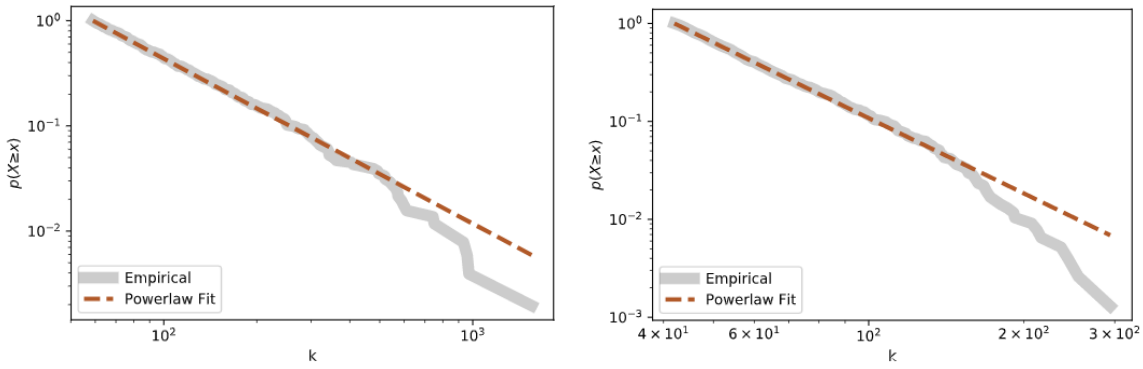
Furthermore, as the degree distribution is approximately follows a power law distribution, so it's a scale-free network. Also after using python function *distribution_compare* to compare power-law fit with exponential distribution, the result shows that power-law is preferred (p=0.0038 is significant).

Using power-law to fit the total degrees is shown in Figure 4. The coefficients are $\gamma = 2.861, \gamma_{in} = 2.586, \gamma_{out} = 3.626$ and in this network, $k_{min}^{in} = 59, k_{min}^{out} = 42, k_{min} = 98$. According to $k_{max} = k_{min} N^{\frac{1}{\gamma - 1}}$, $k_{max}^{in} = 12017, k_{max}^{out} = 1041, k_{max} = 9099$. Here $k_{min}(k_{max})$ is the smallest (largest) degree for which the power law fit holds.



(a) (b)

Figure 3: (a) In degree distribution (b) Out degree distribution



(a) (b)

Figure 4: (a) In power law fit (b) Out power law fit

3.5 Communities Detection

The partition using Greedy Algorithm found consists of 10 communities. The modularity of this partition is 0.2957. The partition using *Fast unfolding of communities in large networks* in *Gephi* with default resolution in found consists of 7 communities (values 0 to 6). The modularity of this partition is 0.368. Then *Gephi* are chosen to analyze and visualize the network respectively.

3.6 Attack Thresholds

This network has scale-free property. Then we regard it as an undirected network during analysis of the attack tolerance. Here we have calculated the coefficient $\gamma = 2.861$ and $k_{min} = 98, k_{max} = 9099$. Take maximum degree of 1845.

Since the robustness of scale-free network is breakdown threshold

$$f_c = 1 - \frac{1}{\frac{\gamma-2}{3-\gamma} k_{min}^{\gamma-2} k_{max}^{3-\gamma} - 1} \sim 0.99999$$

This network is difficult to be broken apart under random failures because of large hubs.

4 Further Discussions

4.1 Classification and Visualization

After the partition, nodes belong to 7 communities mainly. In each community, we choose the nodes such that their in degrees and out degrees are larger than specific given threshold (normally 15). This is the filter of hubs in the network. From the example figures (5 and 6), types of nodes in each community can be identified due to the labels.

1. Natural elements and matters related to chemistry and physics such as soil and oxygen (12.96% nodes)

2. North American cities, histories and music such as The United States (14.55% nodes)

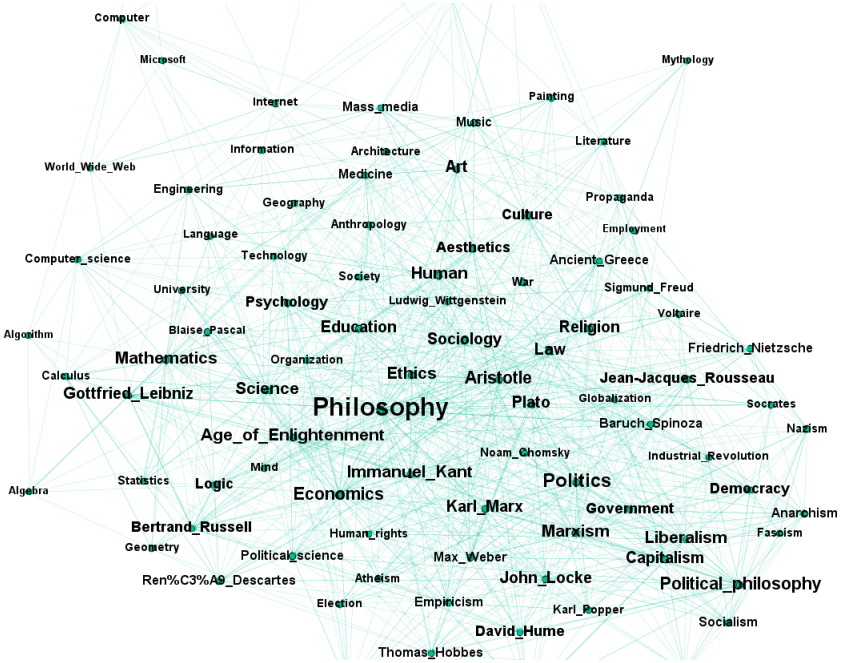


Figure 5: Classifications of subjects nodes

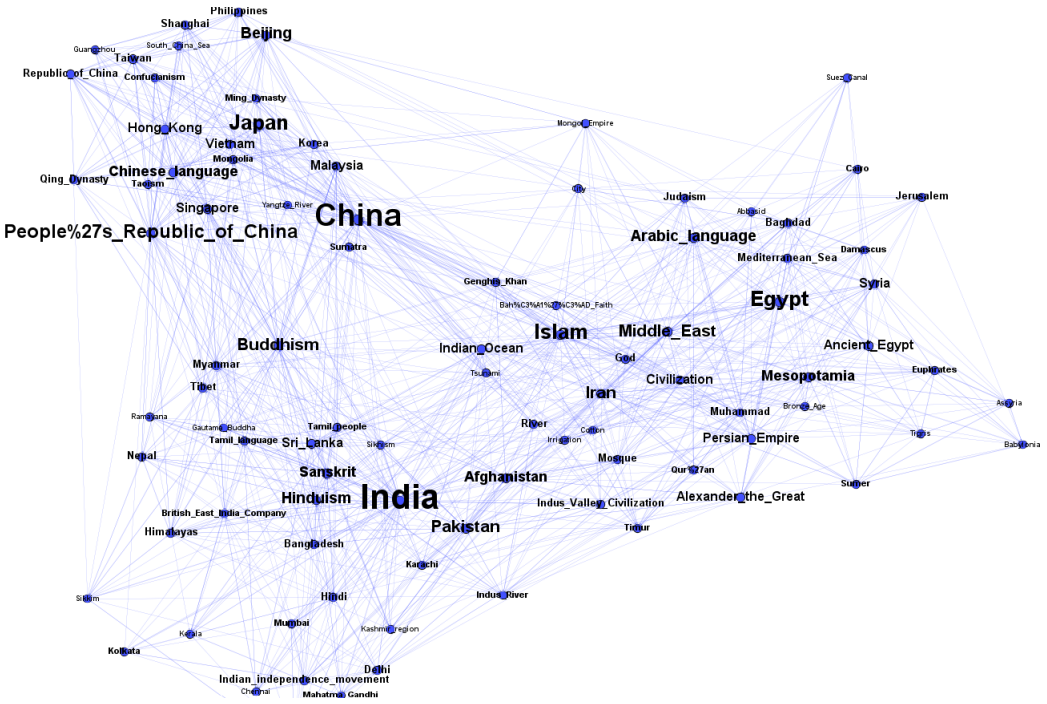


Figure 6: Asia, Africa countries nodes

3. Global countries, cities and languages they use (14.92% nodes)
4. Academic subjects, science and technologies such as mathematics and Internet (12.5% nodes)
5. Classifications of scientific concepts such as biology and geography (17.27% nodes)
6. European history of cultures and countries such as Roman Empire (17.77% nodes)
7. Africa, Asia such as India, China, Egypt (9.02% nodes)

The community detection algorithms has its limitation. There are some actual distinctions between some nodes and our subjective classifications because one node could have multiple characteristics of different communities. For example, node Europe belongs to list of countries' names as well as the European history part. Language belongs to academic subject and global countries.

Furthermore, during the partition, some unrelated categories are combined together such as music and North America. Nodes *musical instrument*, *guitar* and *Johann_Sebastian_Bach* connect large number of other nodes in other communities, but in their own community, they are only ordinary nodes.

4.2 Hubs

Since the analysis in last subsection shows that the hubs in each community may not be the hubs in the whole network. Therefore we use the whole network to find out which nodes are significant in in-degree and out-degree. Set the filter of in-degree 40 and out-degree 30. We finally select 588 (12.8%) potential hubs that connect more with others. There are 23040 (19.22%) edges between them.

4.3 Communities' distance

After we separate the nodes into seven communities with each one has its corresponding type we defined, we have little idea about the contact between each community. In order to

Partition	Number of Hubs	Examples
1	84	'Isaac_Newton', 'Steel', 'Coal'
2	54	'United_States_dollar', 'Canada'
3	187	'Barbados', 'Morocco', 'North_Africa'
4	46	'Literature', 'Politics', 'Religion'
5	55	'Fruit', 'Mammal', 'Cattle'
6	94	'England', 'Viking', '11th_century'
7	65	'Egypt', 'Japan'

Table 1: Hubs in 7 partitions

	1	2	3	4	5	6	7
1	-	2.3999	2.3758	2.3279	2.3043	2.4486	2.2954
2	2.5106	-	2.0772	2.1981	2.3249	2.2094	2.0926
3	2.4537	2.1061	-	2.1360	2.2723	2.1060	1.9853
4	2.4117	2.2951	2.1988	-	2.4020	2.2357	2.1652
5	2.3080	2.2744	2.1735	2.2518	-	2.3190	2.1469
6	2.5325	2.2811	2.1236	2.1660	2.4110	-	2.1255
7	2.4293	2.1877	2.0436	2.1388	2.2898	2.1828	-

Table 2: Distance Between 7 Communities

find the inner link between different types of communities and see if some communities are closer than others, we then try to find the average distance between communities.

Our approach is, use the hubs we defined above, and the extended material about a matrix contains the shortest-path distances between all pairs of articles, to compute the distance. This is reasonable as many small degree articles just connect to their own community's nodes, so the most general case to reach a point from another community is to pass through hubs.

After calculation, the result shows in Table 4.3. We can see that the distance between the two communities are ranged from 1.9853 to 2.4486, which means the community is distributed balanced. As all 588 hubs are connected and the maximum length of the path between hubs is 4, so we can draw a conclusion that hubs are closely linked with each other, thus one possible path from one community to another can by connecting the hubs and going directly without passing other community.

4.4 Comparison between network hubs and hubs that are clicked more often

When playing the game of Wikispeedia, a human player tends to choose points that he is more familiar with, but these points may not necessarily be the hubs in the network [4]. We are curious about how different the points that people click on when choosing a path are different from the points with a larger degree in the network itself.

First, based on the previous analysis, we selected points in the network that have a relatively large out-degree and in-degree. We believe that such points are hubs with a large degree in the network. Then, we counted the number of occurrences of points on the way of all successful paths in the data set, and ranked these points in descending order of the number of occurrences, and regarded the points with the highest number of occurrences as the points that people clicked more frequently.

Point Names	Number of Clicks
United States	8852
Europe	4336
United Kingdom	3888
England	3234
Earth	3191
Africa	2721
World War II	2275
North America	1851
Germany	1725
Animal	1694

Table 3: top ten most clicked points

We have selected a total of 588 hubs with high degrees and 588 points that have been manually clicked. According to the comparison, a total of 348 points overlap. In other words, the coincidence rate of points with a high degree and high manual click rate is as high as 59.1837%. So we can conclude that the overlap between the more clicked points and the higher degree points in the network is quite high.

4.5 Recommended paths

Because of the high coincidence rate between the points with high degrees and the points with high manual click-through rates, it means that many players who successfully walk from the starting point to the end have passed the hubs on the way, so passing the hubs can have a greater probability of success. For players who are stuck in the wikispeedia games, we propose the following algorithm to give players tips, to help players solve the game faster.

STEP 1: Recommend to players the hubs that are close to the starting point.

STEP 2: Through the close connection between hubs, the player can go from the hub near the start point to the hub near the endpoint.

STEP 3: Let the player go directly to the endpoint from the hub that is close to the destination.

It can be seen from the Degree Correlation Matrix of the network that all hubs are connected and most of the two hubs can be reached in 2-4 steps. This conclusion provides great theoretical support for the algorithm we provide.

This algorithm for finding the path between the starting point and the ending point is not necessarily the fastest, but it must be very practical. Compared to finding the shortest path, it is feasible for humans to quickly move from the starting point to the endpoint through the connection between hubs.

5 Conclusion

In this article, we analyze the characteristics and structure of the Wikipedia network based on wikispeedia data. This part includes degree distribution, connectedness, scale-free property, community, attack thresholds, etc. Then, we further discuss the hubs of the network, the comparison of hubs, and the distance between communities. Finally, we propose a feasible hint algorithm for human players who play the game ”wikispeedia”.

References

[1] SNAP:Web data:. Wikispeedia navigation paths.

[2] Shyong K Lam and John Riedl. The past, present, and future of wikipedia. *Computer*, 44(3):87–90, 2011.

[3] Jeremy Neiman. What wikipedia' s network structure can tell us about culture.

[4] Robert West and Jure Leskovec. Human wayfinding in information networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 619–628, 2012.

[5] Robert West, Joelle Pineau, and Doina Precup. Wikispeedia: An online game for inferring semantic distances between concepts. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

A Codes

```
# %%
# import the network
g= pd.read_csv(r"D:\\Grade 3\\Network\\HandsOnCode\\wikilinks.tsv",sep="\t",
                                        names=["Source","Target"],header=0)
g=nx.from_pandas_edgelist(g, 'Source', 'Target', create_using=nx.DiGraph())
# %%
degrees=[g.degree(node) for node in g]
in_degrees = [g.in_degree(node) for node in g]
out_degrees=[g.out_degree(node) for node in g]
# %%
# filter
hubs=[node for node in g
if (g.in_degree(node)>=40)&(g.out_degree(node)>=30)]
pd.DataFrame(hubs).to_csv("hubs4030.csv")
# %%
# read from Gephi
```

```
partition2=pd.read_csv("allnodes.csv")
partition2=dict(zip(partition2['Id'],partition2['modularity_class']))
# %%
print([node for node in g
if (partition2[node] == 6)&(node in hubs)])



# processing matrix types
with open("shortest-path-distance-matrix.txt", "r") as f:
    data = f.read()
littleStr = []
for i in range(len(data)):
    littleStr.append(data[i:i+1])
m = -1
for i in range(4604):
    for j in range(4605):
        m = m + 1
        if(littleStr[m] != '\n'):
            list_three[i][j] = littleStr[m]
list_three = pd.DataFrame(list_three)

# indegree & outdegree calculation
counts=list()
s=list()
a = 'aa'
count = 0
for i in range(len(links)):
    if(a != links.iloc[i,0]):
        if(count != 0):
            counts.append(count)
            s.append(links.iloc[(i-1),0])
            count = 0
        a = links.iloc[i,0]
    count = count + 1
```

```
counts = pd.DataFrame(counts)
counts.columns = ['outdegree']

# find the hub list
hublist = list()
numlist = list()
for i in range(len(article3)):
    if(article3.iloc[i,2] >= 30):
        if(article3.iloc[i,3] >= 40):
            hublist.append(article3.iloc[i,0])
            numlist.append(article3.iloc[i,1])
finaldata = pd.merge(hublist,partition, on = 'node', how='left')
parnum = finaldata.groupby("partitions").size()
parnum

# calculate the distance
mat = zeros((8,8))
for k in range(8):
    for l in range(8):
        m = zeros((parnum[k],parnum[l]))
        for i in range(parnum[k]):
            for j in range(parnum[l]):
                m[i][j] = z.iloc[finaldata[(finaldata['partitions'] == k)].iloc
                                                        [i,2],finaldata[(
                                                        finaldata['partitions']
                                                         == l)].iloc[j,2]]
                mat[k][l] =  m.mean()
```