

NOVEMBER 2023

---

# EXERCISE 3

---

M.Amin HosseinNiya

Presented to:  
Dr. Teymourpour



# ایجاد شبکه‌ی هم‌آیی کلمات کلیدی برای دکتر امیر البدوی

شبکه هم‌آیی کلمات کلیدی co-word برای پایان نامه ها و مقالات استاد کد خود را استخراج کنید و بسازید. نام استاد را هم ذکر کنید. شبکه را در igraph خوانده و حداقل دو مرکزیت آن را حساب کنید. نمودار همبستگی scatter plot این دو مرکزیت را برای مشاهدات (که همان گره‌های شبکه هستند) بکشید. شبکه را با اندازه گره ها در خود igraph ترسیم کنید (با برچسب فارسی). کدام گره ها مهمترند؟ شبکه مفاهیم استاد را تفسیر کنید.

ابتدا شبکه‌ی هم‌آیی کلمات کلیدی را استخراج کردم. به‌گونه‌ای که هر دو تگی که در یک پایان‌نامه آمده‌اند، در این شبکه به هم متصل هستند:

```
## To read the json file and it's context:
```

```
import json
```

```
with open("output3.json", "r", encoding='utf-8') as file:
```

```
    data = json.load(file)
```

```
file.close()
```

```
data_copy = data.copy
```

```
for item in data:
```

```
    if "tags" in item.keys():
```

```
        item["tags"] = [tag.strip("12345").replace("\u200c", " ") for tag in
```

```
item["tags"]]
```

```
all_tags = list()
```

```
for item in data:
```

```
    if "tags" in item.keys():
```

```
        [all_tags.append(tag) for tag in item["tags"]]
```

```
all_titles = list()
```

```
for item in data:
```

```
    all_titles.append(item["title"])
```

# ایجاد شبکه‌ی هم‌آیی کلمات کلیدی برای دکتر امیر البدوی

```
len(all_titles)
all_tags = [element.strip("12345()") for element in all_tags]
len(all_tags)
unique_tags = list(set(all_tags))
len(unique_tags)
unique_tags
# unique_tags = [i.replace("\u200c", " ") for i in unique_tags]
# unique_tags
# all_tags = [i.replace("\u200c", " ") for i in all_tags]
len(all_tags)
import numpy as np
adjacency_matrix = np.zeros((len(unique_tags), len(unique_tags)))
tags_indices = {word:index for index, word in enumerate(unique_tags)}
## To create Adjacency Matrix:
for item in data:
    if "tags" in item.keys():
        for i in range(len(item["tags"])):
            for j in range(i+1, len(item["tags"])):
                tag1 = item["tags"][i]
                tag2 = item["tags"][j]
                index1 = tags_indices[tag1]
                index2 = tags_indices[tag2]
                adjacency_matrix[index1][index2] = 1
                adjacency_matrix[index2][index1] = 1
```

سپس دو مرکزیت درجه و پیج‌رنک را محاسبه کردم:

```
from igraph import Graph
from igraph import plot
graph = Graph.Read_Adjacency("adjacency_matrix.txt")
graph.vs["name"] = reshaped_unique_tags
graph.vs["label"] = reshaped_unique_tags
degree_centrality = graph.degree()
pagerank_centrality = graph.pagerank()
```

## ترسیم Scatter Plot و گراف

---

اسکترپلات این دو مرکزیت و همچنین گراف شبکه‌ی ایجادشده را رسم کردم:

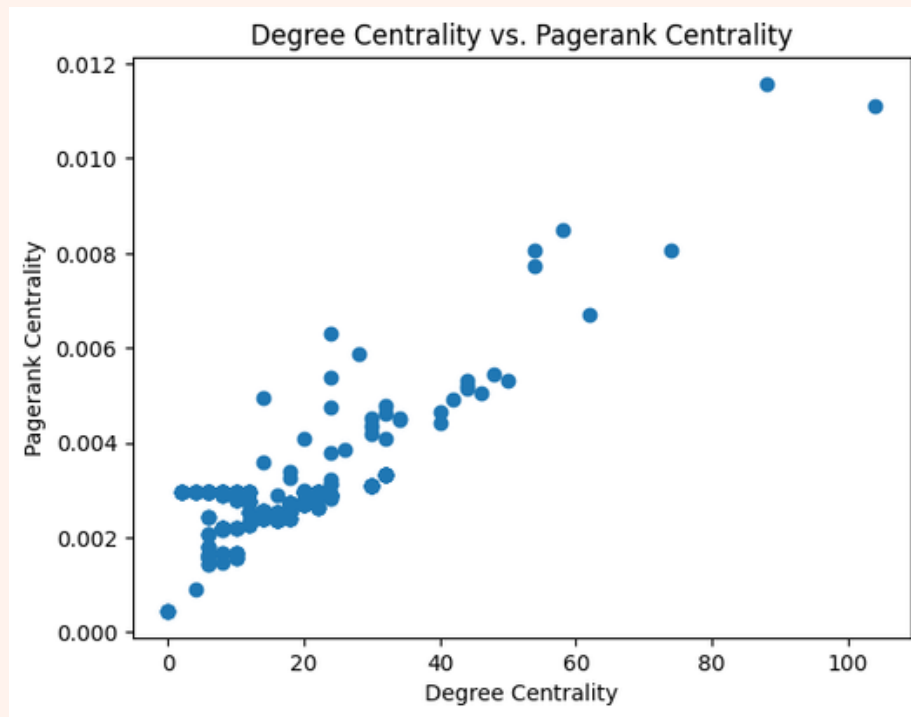
```
import matplotlib.pyplot as plt
plt.scatter(degree centrality, pagerank centrality)
plt.title("Degree Centrality vs. Pagerank Centrality")
plt.xlabel("Degree Centrality")
plt.ylabel("Pagerank Centrality")
plt.show()
import arabic_resaper

min_size = 10
max_size = 100
node_sizes = [(degree - min(degree centrality)) / (max(degree centrality) -
min(degree centrality))
               * (max_size - min_size) + min_size for degree in degree centrality]

layout = graph.layout("kk") # Choose a layout algorithm (e.g., "circle", "fr", "kk",
etc.)
# visual_style = {}
# visual_style['vertex.label.family'] = 'Arial'
# visual_style['vertex.label.cex'] = 1.2
plot(graph, layout=layout, vertex_size=node_sizes, vertex_color="lightblue",
edge_color="gray",
     vertex_frame_color="black", edge_width=1.0, bbox=(1200, 300), margin=20)
```

# ترسیم Scatter Plot و گراف

اسکترپلات:



گراف:

