

NOVEMBER 2023

EXERCISE 4

M.Amin HosseinNiya

Presented to:
Dr. Teymourpour



استخراج گراف و ترسیم در Gephi

ابتدا فایل جیسون شامل اطلاعات مربوط به مقالات و پایان‌نامه‌های دکتر **امیر البدوی** را در پایتون خواندم. سپس به انجام پیش‌پردازش‌های لازم برای تشکیل ماتریس مجاورت پرداختم:

```
## 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()") 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"])
```

```
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)
```

```
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)}
```

استخراج گراف و ترسیم در Gephi

ماتریس مجاورت را به گونه‌ای تشکیل دادم که هرگاه دو تگ همزمان در یک مقاله آورده شده‌اند، مقدار درایه‌ی با سطر و ستون نظیر آن دو تگ برابر یک شود. سپس این گراف را با جزئیات لازم برای خوانده‌شدن مناسب در Gephi (نظیر لیبل مناسب، مرکزیت درجه و...) استخراج کردم:

```
## 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
```

```
for item in data:
```

```
    for tag in item["tags"]:
```

```
        print(unique_tags.index(tag))
```

```
file_path = 'adjacency_matrix.txt'
```

```
np.savetxt(file_path, adjacency_matrix, fmt='%d')
```

```
## To export the graph:
```

```
from igraph import Graph
```

```
from igraph import plot
```

```
graph = Graph.Read_Adjacency("adjacency_matrix.txt")
```

```
graph.vs["name"] = unique_tags
```

```
graph.vs["label"] = unique_tags
```

```
graph.vs["label"]
```

```
layout = graph.layout("kk")
```

```
visual_style = {"vertex_color": "blue", "vertex_size": 20, "vertex_label": graph.vs["name"],  
               "edge_arrow_size": 0.5}
```

```
degree_centrality = graph.degree()
```

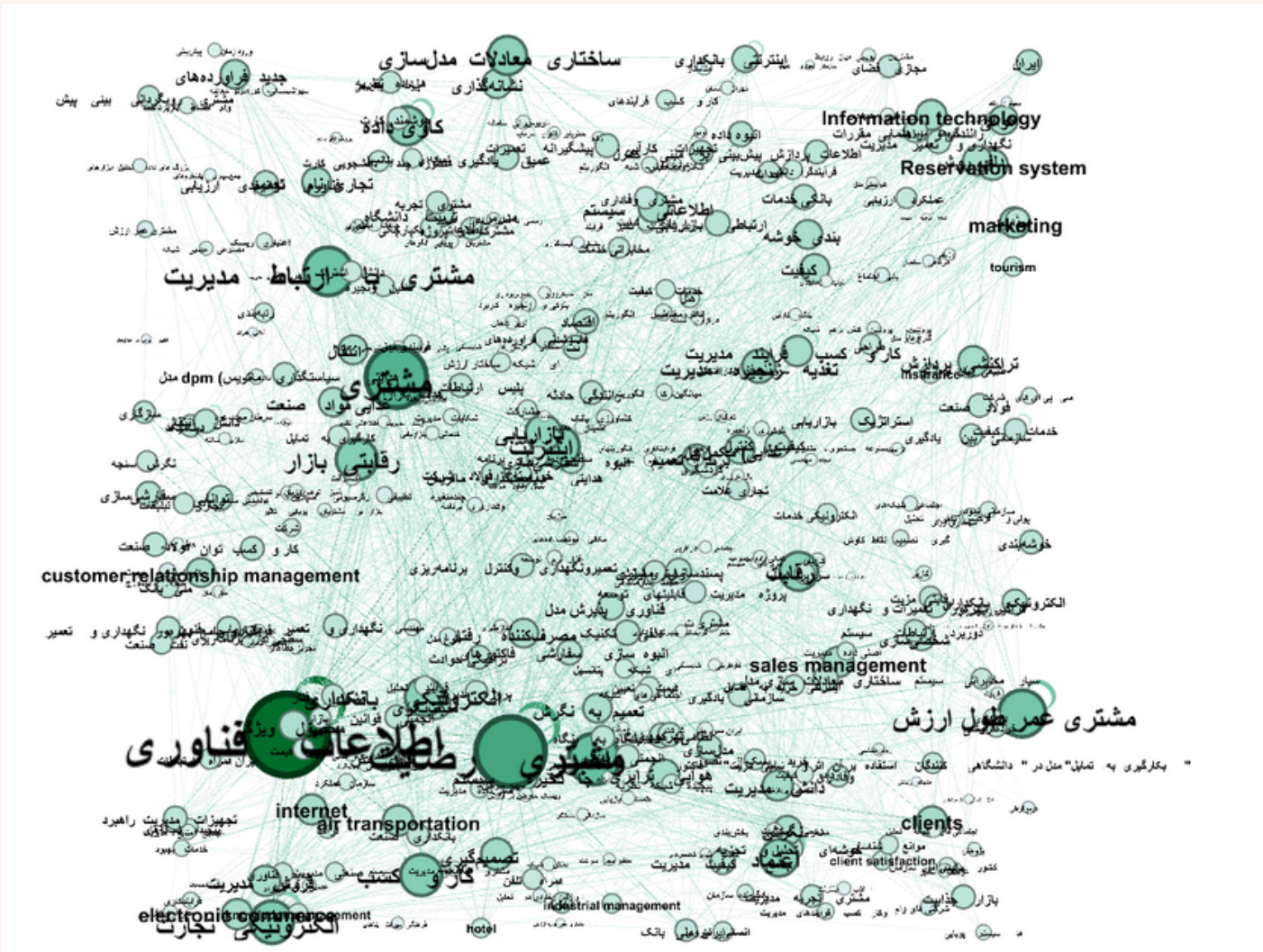
```
graph.vs["degree_centrality"] = degree_centrality
```

```
file_path = "graph.graphml"
```

```
graph.write_graphml(file_path)
```

استخراج گراف و ترسیم در Gephi

پس از آن گراف را در Gephi ایمپورت کردم و آن را نمایش دادم. نمایش لیبل‌ها را فعال کردم. همچنین اندازه‌ی گره‌ها متناسب با مرکزیت درجه‌ی آن‌ها تنظیم شد:



مشخص است که گره‌های فناوری اطلاعات، رضایت مشتری، مشتری، ارزش طول عمر مشتری و مدیریت ارتباط با مشتری مرکزیت درجه‌ی بیشتری دارند و لذا مهم‌تر هستند.

ترسیم شبکه‌ی دوبخشی پایان‌نامه-واژه‌ی کلیدی

ابتدا گراف مربوط به این بخش را با فرمت edge-list تهیه کردم. سپس نودهای این گراف را به دو لایه‌ی مجزا تفکیک کردم تا ترسیم آن‌ها با sugiyama layout امکان‌پذیر شود.

```
## To draw the Bipartite Graph of papers and their tags:
# title_indices = {title:index for index,title in enumerate(all_titles)}
edge_list = np.zeros((len(all_tags), 2))
counter = 0
for index, item in enumerate(data):
    if "tags" in item.keys():
        for tag in item["tags"]:
            edge_list[counter][0] = index
            edge_list[counter][1] = tags_indices[tag]+1000
            counter += 1
edge_list = edge_list.astype("int32")
edge_list
file_path = 'bipartite_garph.txt'
np.savetxt(file_path, edge_list, fmt='%d')
##
bipartite_garph = Graph.Read_Edgelist("bipartite_garph.txt", directed=False)
layer_assignments = np.zeros(1400)
with open("bipartite_garph.txt", "r") as file:
    max = 0
    for line in file:
        layer_assignments[int(line.split()[0])] = 0
        layer_assignments[int(line.split()[1])] = 1
        # line = [int(element) for element in line.split()]

layer_assignments = layer_assignments.astype("int32")
layer_assignments = layer_assignments.tolist()
bipartite_garph.vs["layer"] = layer_assignments
layout = bipartite_garph.layout_sugiyama(layers=bipartite_garph.vs["layer"])
# layout = bipartite_garph.layout_kamada_kawai()
visual_style = {"vertex_color": "blue", "vertex_size": 20, "vertex_label": graph.vs["name"],
"edge_arrow_size": 0.5}
plot(bipartite_garph, layout=layout, bbox=(700, 200), **visual_style)
```

ترسیم شبکه‌ی دوبخشی پایان‌نامه-واژه‌ی کلیدی

خروجی به شکل زیر بود:

