import networkx as nx

import pandas as pd

# 创建一个空的无向图

G = nx.Graph()

# 从Excel文件加载数据

excel\_file = "CADVis\_Author.xlsx" # 替换为您的Excel文件路径

df = pd.read\_excel(excel\_file)

# 根据相同的 work\_id 创建节点和边

for work\_id, group in df.groupby("work\_id"):

authors = group["Fullname"].tolist()

order = group["Order"].tolist()

# 添加节点（作者）

for author in authors:

G.add\_node(author) # 添加节点（作者）

# 添加带有作者顺序的边（合作关系）

for i in range(len(authors)):

for j in range(i + 1, len(authors)):

G.add\_edge(authors[i], authors[j], order=order[i]) # 使用order[i]来表示作者顺序

# 创建一个子图，只包括与Marc Aurel Schnabel相关的节点和边

subgraph = G.subgraph(["Marc Aurel Schnabel"] + list(G.neighbors("Marc Aurel Schnabel")))

# 计算Marc Aurel Schnabel的度中心性、介数中心性和特征向量中心性

degree\_centrality = nx.degree\_centrality(subgraph)

betweenness\_centrality = nx.betweenness\_centrality(subgraph)

eigenvector\_centrality = nx.eigenvector\_centrality(subgraph)

# 创建一个 DataFrame 存储中心性结果

centrality\_df = pd.DataFrame({

"Author": list(subgraph.nodes()),

"Degree Centrality": [degree\_centrality[author] for author in subgraph.nodes()],

"Betweenness Centrality": [betweenness\_centrality[author] for author in subgraph.nodes()],

"Eigenvector Centrality": [eigenvector\_centrality[author] for author in subgraph.nodes()]

})

# 计算Marc Aurel Schnabel的共同作者

marc\_aurel\_schnabel\_collaborators = set(subgraph.neighbors("Marc Aurel Schnabel"))

# 输出Marc Aurel Schnabel的共同作者数量

print(f"Marc Aurel Schnabel与{len(marc\_aurel\_schnabel\_collaborators)}位共同作者合作。")

# 保存中心性度量的结果到Excel文件

centrality\_df.to\_excel("marc\_aurel\_schnabel\_centrality\_results.xlsx", index=False)

# 打印前几行以查看结果

print(centrality\_df.head())

Marc Aurel Schnabel与58位共同作者合作。

Author ... Eigenvector Centrality

0 Thomas Kvan ... 0.030480

1 Davide Pierini ... 0.034945

2 Steve Kuan ... 0.026980

3 Weidong Li ... 0.026980

4 Mitra Homolja ... 0.024708