Прикладные задачи

Изучение проблемных областей исследований посредством представления фактов и знаний на графах

Дано:

Имеется база данных публикаций биомедицинской литературы PubMed (> 30 млн. article).

Имеются программные интерфейсы доступа к ней.

Требуется:

1. Идентифицировать всех авторов и их принадлежность к учреждениям проводящим исследования (по какому-либо произвольно формируемому запросу – напр. “Decision Support Systems”)
2. Провести сортировку публикаций по значимости (квантиль) издания
3. Построить граф связности авторов – (узлы-авторы, ноды-количество совместных публикаций)

Пример решения:

import csv  
import itertools  
  
from pymed import PubMed  
  
pubmed = PubMed(tool="My Graph Analysis", email="artem@email.address")

# поисковый запрос

query = "decision support system[Title]"  
print("Найдено", pubmed.getTotalResultsCount(query))  
  
# Execute the query against the API  
results = list(pubmed.query(query, max\_results=1000))  
  
# создание узла для каждого уникального автора публикации  
nodes = {  
 author: index  
 for index, author in enumerate(  
 set(  
 itertools.chain.from\_iterable(  
 [  
 [  
 f'{author["lastname"]} {author["firstname"]}'  
 for author in article.authors  
 ]  
 for article in results  
 ]  
 )  
 )  
 )  
}  
  
# Создание ребра графа для каждой комбинации автор/со-автор

edges = list(  
 itertools.chain.from\_iterable(  
 [  
 [combination for combination in itertools.combinations(co\_author\_list, 2)]  
 for co\_author\_list in [  
 [  
 nodes[f'{author["lastname"]} {author["firstname"]}']  
 for author in article.authors  
 ]  
 for article in results  
 ]  
 ]  
 )  
)  
  
# Замена дублей в списке на веса

edges = set([(edge[0], edge[1], edges.count(edge)) for edge in edges])  
  
  
# файл для узлов  
with open("./nodes.csv", "w", encoding="utf8", newline="") as nodes\_file:  
  
 # CSV writer  
 writer = csv.writer(nodes\_file, delimiter=",")  
  
 # пишем header  
 writer.writerow(["id", "label"])  
  
 # цикл по авторам и создание строк   
 for name, index in nodes.items():  
 writer.writerow([index, name])  
  
# создание файла с ребрами  
with open("./edges.csv", "w", encoding="utf8", newline="") as edge\_file:  
  
 # создаем CSV writer  
 writer = csv.writer(edge\_file, delimiter=",")  
  
 # пишем header  
 writer.writerow(["source", "target", "weight"])  
  
 # в цикле пишем ребра(дуги) графа  
 for edge in edges:  
 writer.writerow(edge)

Для построения графа используем Gephi.