Primo Progetto di Social Computing

A.A. 2023/24 –14 Novembre 2023

Introduzione

Il progetto si concentra sulla creazione e analisi di reti di coautorialità tra ricercatori utilizzando dati estratti da Google Scholar (https://scholar.google.com/). L'obiettivo è quello di integrare un file CSV fornito con informazioni dettagliate sugli autori, e quindi analizzare le connessioni di coautorialità. Verrà utilizzato Python con le librerie SerpAPI per il recupero di dati e Pandas per la loro manipolazione. Successivamente, le reti sociali prodotte verranno visualizzate e analizzate con NetworkX.

Specifiche

Assieme a questo pdf trovate il file **nodes.csv** . Tale file è strutturato come in figura, e contiene il nominativo e l'affiliazione di 7 ricercatori.

name	**affiliations**
David La Barbera	Università Degli Studi Di Udine
Michael Soprano	Postdoctoral Research Fellow at the University of Udine
Kevin Roitero	University of Udine
Stefano Mizzaro	Full professor of Computer Science and Information Technology
Damiano Spina	School of Computing Technologies, RMIT University
Gianluca Demartini	Associate Professor at the University of Queensland
Eddy Maddalena	Università degli Studi di Udine, Italy

1. A partire da **nodes.csv**, utilizzare la libreria Python SerpAPI per scaricare, per ciascuno dei 7 autori elencati:
   - **author_id**: ID identificativo del profilo Google Scholar
   - **cited_by**: numero totale di citazioni ricevute
   - **interests**: elenco degli interessi di ciascun autore

   Sfruttando la libreria Python Pandas, usare la struttura dati DataFrame per aggiornare il file originale con apposite colonne e memorizzarlo nella cartella /**data**.

2. Per ciascuno dei 7 autori, utilizzare il suo ID per accedere al relativo profilo Google Scholar e scaricare l'elenco dei suoi coautori, sempre via SerpAPI. Con tale elenco di nomi:
   a. Utilizzare le SerpAPI per cercare su Google Scholar un ricercatore che corrisponde a tale nome. Per ciascuno, salvare **name**, **affiliations**, **author_id**, **cited_by** e **interests** in un nuovo DataFrame contenente tutte queste informazioni relative ai coautori dei 7 autori originari.

non avendo l'affiliazione nei co-authors prendo il primo della lista, trovo un solo profilo che li rappresenta (solo su profiles, non su authors)

b. Concatenare il DataFrame con i 7 autori originari e quello dei coautori generato al punto 2a in un unico DataFrame.
NOTA BENE: è sufficiente effettuare la ricerca dei profili per nome, non accedere al loro profilo tramite id. eliminare i duplicati
ASSUNZIONE: in questo caso non potete identificare il profilo corretto tramite il valore di affiliations, quindi assumete che quello corretto sia il primo ritornato nella lista di authors.
c. Creare un terzo DataFrame con le colonne **author1**, **author2** che rappresenta le co-authorship. In tale DataFrame, una riga rappresenta un arco di coauthorship tra due autori.
ESEMPIO: David La Barbera, Michael Soprano è una riga del DataFrame creato al punto 2c se Michael Soprano è coautore di David La Barbera. La co-authorship è binaria, non pesata.

A questo punto avrete prodotto due DataFrame:
- Uno per le informazioni relative agli autori (originali + i relativi coautori) e contenente per ciascuno di essi i valori di **name, affiliations, cited_by, interests**. Salvare come **nodes.csv** nella cartella **/data** tale DataFrame.
- Uno per le relazioni di co-authorship dai 7 autori principali verso i relativi coautori con colonne **author1, author2**. Salvare come **edges.csv** nella cartella **/data** tale DataFrame.

3. Utilizzando i due DataFrame prodotti:
   a. Generare un grafo indiretto che ne rappresenta le informazioni contenute. Salvare (serializzare) il grafo in locale nella cartella **/graphs** .
   b. Visualizzare il grafo prodotto colorando i nodi a seconda del loro grado con le seguenti colorazioni: grigio per nodi con grado uguale a 1, blu per nodi con grado compreso tra 2 e 10, viola per nodi con grado compreso tra 11 e 20, giallo per nodi con grado maggiore di 20. Inoltre, per ciascun nodo visualizzare il nome dell'autore. Salvare la visualizzazione nella cartella **/visualizations**.

4. A partire dal grafo prodotto al punto 3:
   a. Generarne un secondo dove il numero di nodi è lo stesso, mentre il numero di archi è aumentato di 50 utilizzando la tecnica del Preferential Attachment. Salvare il grafo in locale nella cartella **/graphs**.
   b. Visualizzare e salvare il nuovo grafo come fatto al punto 3b.

5. Per entrambi i grafi prodotti calcolare:
   a. Coefficiente di clustering medio
   b. Centro del grafo
   c. Raggio
   d. Distanza Media
   e. Transitività
   f. Coefficienti Omega e Sigma, per stimare la "small-world-ness"
   g. Riassumere le informazioni in un DataFrame, dove ogni riga rappresenta le informazioni relative ad un grafo, ed ogni colonna le informazioni relative ad una misura calcolata per quel grafo. Salvare nella cartella **/results** tale DataFrame.

6. Per entrambi i grafi prodotti calcolare per ogni nodo:
   a. Degree Centrality
   b. Betweenness Centrality
   c. Closeness Centrality
   d. Pagerank

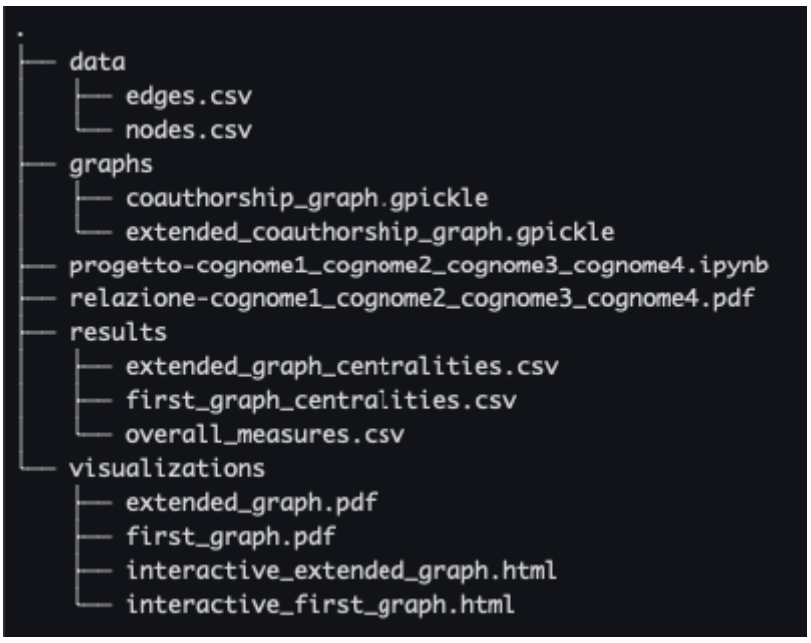
e. HITS, per calcolare i valori di hubness e authority
f. Riassumere le due informazioni in un DataFrame per ciascun grafo, dove ogni riga rappresenta le informazioni relative ad un nodo, ed ogni colonna le informazioni relative ad una misura calcolata per quel nodo. Salvare nella cartella **/results** tali DataFrame.

7. Produrre una visualizzazione interattiva con PyVis dei due grafi con colorazioni di nodi ed archi a piacimento. Salvare i due output in formato HTML nella cartella **/visualizations** .

Come consegnare

1. Dovete costituire dei gruppi che **devono** essere formati da **quattro** persone (i gruppi più o meno numerosi verranno penalizzati)

2. Vanno consegnati i seguenti elementi, contenuti in una cartella zippata denominata **cognome1_cognome2_cognome3_cognome4** e strutturata come in figura:



- **data**: contiene nodes.csv, edges.csv
- **graphs**: contiene le serializzazioni dei grafi prodotti ai punti 3 e 4.
- **results**: contiene tutti i dataframe prodotti per calcolare le misure richieste (uno come output del punto 5, due, uno per grafo, come output del punto 6).
- **visualizations**: contiene le visualizzazioni prodotte al punto 3, 4 e 7.

- **relazione-cognome1_cognome2_cognome3_cognome4.pdf :** La relazione che descrive tutto il lavoro svolto è di al massimo 5 pagine. Va consegnata in formato **PDF** e strutturata come da esempio in appendice al presente documento.
- **progetto-cognome1_cognome2_cognome3_cognome4.ipynb**

3. Consegnate via mail a entrambi i docenti (un unico messaggio indirizzato a entrambi)
    - mizzaro@uniud.it
    - david.labarbera@uniud.it
    - oggetto della mail nel formato:
      [Progetto SocCom 1] cognome1_cognome2_cognome3_cognome4
    - in allegato un unico file zippato che produce una cartella con nome cognome1_cognome2_cognome3_cognome4

4. **Scadenza: Domenica 10 Dicembre 2023 AoE Timezone**

5. Punteggio:
    - 5 punti in trentesimi per, indicativamente, i migliori 20%,
    - 4 punti per i seguenti 20%,
    - 3 punti per i seguenti 20%,
    - 2 punti seguenti 20%,
    - 1 punto per i seguenti 20%,
    - 0 punti per progetti non adeguati (a discrezione dei docenti) o per chi non consegna

Informazioni aggiuntive

- Viste le limitazione poste sugli endpoint da SerpAPI, fate attenzione a sfruttare la cache per le chiamate già effettuate, come spiegato a lezione (ad es., se fate delle prove scaricate dati che poi vi saranno utili per il progetto, quali i profili dei 7 autori originali). Sfruttate al meglio le 4 key di ciascuno dei membri del gruppo, che sono ampiamente sufficienti per scaricare i dati richiesti.
- Tutte le misure, proprietà e verifiche richieste su grafi sono definite nella documentazione di NetworkX; consultatela per capire come rispondere alle varie richieste
- Scrivete la relazione secondo le linee guida in calce. Adottate lo stile che preferite, purchè rispetti le sezioni specificate e sia consegnata in formato PDF. Usate il software che preferite (LaTex, Word, …).
- Consegnate un notebook adeguatamente commentato e diviso in sezioni rispettando i punti del progetto “dall’alto verso il basso”.
- La valutazione finale è influenzata anche dal codice prodotto e dalla relazione scritta.
- Eventuali extra verranno valutati positivamente solo nel caso in cui il progetto sia stato svolto nella sua interezza.

Titolo: Come scrivere la relazione

Nome1 Cognome1 - matricola - email
Nome2 Cognome2 - matricola - email
Nome3 Cognome3 - matricola - email
Nome4 Cognome4 - matricola - email

1. Executive summary (al più mezza pagina)

Breve riassunto della relazione, che spiega cosa è stato fatto, come è stato fatto, e quali risultati sono stati ottenuti.

2. Metodologia

2.1 Scaricamento Dei Dati

Descrizione delle tecniche adottate per scaricare i dati, assunzioni effettuate, algoritmi implementati.

2.2 Espansione Del Grafo

Descrizione delle tecniche adottate per aumentare il grafo secondo il preferential Attachment

2.3 Costruzione e Visualizzazione dei Grafi

Descrizione delle tecniche adottate per costruire, colorare, e visualizzare i grafi

2.4 Ulteriori Assunzioni

Altre assunzioni che non ricadono nelle precedenti sezioni.

3. Risultati

Sinteticamente e per ciascun punto del progetto (da 1 a 7) presentare gli output calcolati. Commentare le tabelle riassuntive relative alle misure calcolate e le visualizzazioni dei grafi prodotte.

4. Conclusioni (al più mezza pagina)

Al più mezza pagina di considerazioni conclusive (breve riassunto, questioni aperte, eventuali attività extra non richieste, ecc.).