Assignment 01 Date: 30th August, 2019 Last Date of submission: 18th Sept, 2019
Late submission attracts appropriate penalty

Title: Extract a social graph based on co-occurrence and do an analysis of the social graph as per direction provided

Marks : 10 (20 will be scaled down to 10)

Goal:

(a) *To build an **implicit social graph using co-occurrence from a literary text***
(b) *Analyse the graph to correlate your findings with the story*
(c) *Visualize the above social graph using Gephi (your analysis findings and visualizations should correlate)*

Team :

1. It is a group activity
2. Same team has to continue for assignments and project
3. It will be team marking

Marking Guidelines

Marking scheme	**Timeliness (5 marks)**	**Functionality asked for (5 marks)**	**Analysis (5 marks)**	**Visualization (5 marks)**
Max (5)	on time	working code without issue	Now I understand the story better (;-	The visualization reflects key findings in the network
Mid (4)	on time +1- 7 days	working with minor issues	All metrics calculated but correlation with the story is not pointed out.	Visualization done but does NOT reflect any key findings in the network.
Min (3)	delay more than 7 days	working with major issues	All metrics calculated and no analysis really	Poor visualization

Submission procedure:

(a) **Update the teaminformation google sheet in your course folder, update team information and the text chosen for this assignment.**
(b) **Do not copy full assignment from Github to avoid penalty**
(c) Filename for submission will have the format : **A01<Team_XX>.zip**
(d) The zip should contain the following (all files necessary to check your work)
   a. The source text
   b. Any intermediate file (example- list of characters that you used)
   c. The graph data that Gephi imported
   d. Gephi project file (after customization for visualization)
   e. Jupyter notebook with python 3.x code
   f. ***Last part of the jupyter notebook should have 2 brief sections*** i.e.
      i. Comment about your visualization in Gephi. In which way it is correlating with your analysis (point d) ?
      ii. What are your key finding in the analysis (point c) ? Correlate the same with the story.
(e) Mail ZIP file to Bhaskarjyoti01@gmail.com on or before due date

Assignment 01 Date: 30th August, 2019 Last Date of submission: 18th Sept, 2019
Late submission attracts appropriate penalty

Instruction:

(a) **Dataset :** Choose an English text (novel, drama etc) available at Project Gutenberg website.
(b) **Quickly read up about the novel/text** from net and **make a list of all the characters and their aliases, if required. Read up about the story and key points even before you start. You probably will depict few or all of these key points in your analysis/visualization.**
(c) Your will have following steps (also refer to the detailed note) :
   a. **Pre processing the text**. Here there is no fixed guidance as it depends on the text. Feel free to do your own steps (so as to yield a successful analysis) .
   b. **Building adjacency matrix or adjacency list** for the extracted implicit undirected graph using some co-occurrence algorithm.
   c. If required, **process the extracted graph further** by dropping unnecessary nodes, isolates. Export this final graph so that Gephi can import.
   d. **Using Networkx in Python, perform the analysis** and Refer below for detailed instructions.
   e. **Using Gephi do a visualization** (check previous discussion in class)
   f. **Do the final analy**sis in last section of the Jupyter notebook

Detailed note

1. About analysis using networkx

You should do the followings in your python notebook after extracting the adjacency matrix/list of the weighted undirected social graph. Your final analysis can be done probably in a brain storming session in your team using the results obtained below.

(a) How big is the original graph (no of nodes, edges) ? How many isolates you have ?
(b) Actor level analysis
   a. Find the nodes along with frequency of occurrence.
   b. Identify the set with which you will do the analysis and if required, adjust your matrix/list by re-running your code for the identified set of actors. This can happen only with a very large text.
   c. Eliminate isolates as they do not add any value.
   d. Calculate centrality (degree, betweenness, closeness, eigen vector) and make side by side comparison for the nodes identified
   e. Calculate the ego network size of the identified characters
   f. Now analyse the result obtained so far and identify the main protagonists in this text
(c) Edge level analysis – what are the top few edges in terms of betweenness ?
(d) Analyse the overall cohesiveness of the network, diversity of degree etc.
   a. Find the diameter
   b. Find the average shortest path
   c. Find the degree distribution
   d. Find the size of the largest component

   e. Degree assortativity in the graph
   f. Global clustering coefficient of the graph and local clustering coefficient of the main nodes

(e) Find community in the social graph (Louvain and Clique percolation).
(f) If you feel necessary, you can also try to discover sub structure using k-Clique, K -core etc. You can attempt core-periphery analysis as well.
(g) At this point, put together your analysis and update the Gephi visualization so that it correlates with what you found.

2. About co-occurrence method of finding edges between nodes

- Before you do anything, strip the redundant portion from the text (example – author name, publication, text marking for chapter, non English characters, non text, CR/LF etc.)
- Make a list of characters in the text as they appear. You probably may need to process it further i.e. Tom Hogan and Tom are same in the text and so you will make them same. Browse the net to acquire information about the text you have chosen.
- Do not try very long text with innumerable characters in it.
- Split the text into list of words
- Count the total number of words and then choose a window size (this is something that you may try more than once to find the right window size) . It should encompass at least few lines of text. Use your own logic. For a drama, it can be a scene.
- Now you should implement an algorithm whereby you build the adjacency list for a weighted undirected graph with weight signifying the number of occurrences. You can check the accompanying papers.
- Once the above is done, the rest is Python /networkx code (check the shared code).