Information Retrieval Lab, Session 5: Network analysis

In this session:

- You will use textual data to build a network and try out several network analysis methods seen in class

1 Your task

This is a fairly open-ended lab work session in the sense that you have to come up with your own data to build a network and then study it.

1.1 Building your network

The idea is to use a collection of texts of your choice, in order to build a network. Examples of this are, for example:

- Using Twitter messages as nodes in your network and using a text-based similarity measure to add links between them.
- Using news gathered from the web and linking them also using some kind of text-based similarity measure.
- Using recipes as the sources of text and linking recipes based on the ingredients that they use, or the cooking methods.
- Using classical texts from public domain authors e.g. Shakespeare or Jane Austen or Cervantes or all of them.
- Using abstracts from scientific papers.

You can be as creative as you like. Your resulting network should be connected, and have at least 200 nodes.

If your data comes from Twitter you can use libraries e.g. `pattern` in Python. You can also use libraries that help with the network analysis e.g. `networkx` in Python or `igraph`.

In cased you don't find an appealing dataset, you can use the `london tube graph` or the `Nutritional Food Data`, both from `http://kajeka.com/miru/example-data/`. In the first case you only have to load the text file with edges and do an analysis of the network. In the second case you have to create a graph where each node is a kind of food. Then you have to define a function of similarity between foods and build edges among different nodes when the foods are similar enough form the point of view of nutrients.

1.2 Analyzing your network

Now that you have built a network, you should use the tools seen in class to gather information about it. For each network analysis metric (seen in class) that you can reasonably[1] compute, try to interpret the result based on the knowledge you have about the network and how was generated. For instance:

1. Describe the network a little. How many edges and nodes does it have? What is its diameter? And transitivity? And degree distribution? Does it look like a random network? Apply a pagerank on the nodes and, if it is possible, visualize the network with node sizes proportional to their pagerank.
2. Now, use a community detection algorithm of your choice from the list provided. How many nodes does the largest community found contain? Plot the histogram of community sizes. If possible, plot the graph with its communities.
3. Does the result make any sense, given that you know how you created the network?

2 Deliverables

To deliver: You must deliver a brief report (1 or 2 pages) describing your results. You also have to hand in the source code of your implementations.

Procedure: Submit your work through the Racò at `https://raco.fib.upc.edu/` as a single zipped file.

Deadline: Work must be delivered within 2 weeks from the lab session you attend. Late deliveries risk being penalized or not accepted at all. If you anticipate problems with the deadline, tell me as soon as possible.

[1] Here, *reasonably* means that you spend time in the order of a few hours, not days, writing code and executing it on your network.