# **Chapter Eight: Citation Data Analysis And Visualization**

**Introduction**

In this chapter, we will go through the phases we have done for cleaning, visualizing, analysing and exporting data for the application, and before that define the data that we dealt with.

**Data Format and cleaning**

The data was in a disaster format, with more than 4milion lines, and it was not following a certain order, but each line had a specific prefix, to specify what each line represent:

The prefixes were:

#\* paper Title

#@ Authors

#t year

#c publication venue

#index no

#% index of references (could be multiple lines)

#! Abstract

we used regular expression to create each record, with dropping every record that has no authors. We got CSV file with the records we want.

The main disadvantage of using regular expression was the runtime speed, too slow.

The figures below show the method we used:

**Text

Description automatically generated with low confidence**

Figure 8.1: regular expression definitions.

**Graphical user interface, text, application

Description automatically generated**

Figure 8.2: Generating the cleaned data.

**Data visualization and analysis**

* The library was used for visualization and analysis mainly is Networkx in addition to basic libraries such Numpy, Pandas, matplotlib and math.
* The data is ready for representing it as Graph, now with every publication id we create a node, and link it with the publication that it cited from and keep It isolated node if it hasn’t.
* The nodes were added to list, and edge list was created then the graph was created.
* Our network was dense and quite large with exactly: 629814 nodes and 632751 edges.

Basic analysis: we found the strongly connected components, weakly connected components, finding the density and degree centrality, the figure below show our basic analysis and graph creation.Graphical user interface, text, application, email

Description automatically generated

Figure 8.3: graph creation and basic analysis

The degree centrality was applied depending on the degree value, other degree centrality measurement algorithms were not applicable due to the large data, and the limited resources we have.

We visualized degree distribution then and found that most publications were neither referenced nor cited from other publications, the figures below show the degree analysis and visualization.

**Graphical user interface, text, application

Description automatically generated**

Figure 8.4: in degrees and out degree manipulation.

**A picture containing graphical user interface

Description automatically generated**

Figure 8.5: Degree histogram.

**Chart, line chart

Description automatically generated**

Figure 8.6: in Degree vs out degree histogram.

• Community detection and clustering algorithms: we applied different clustering algorithms, but the average clustering coefficient was nearly zero which means there are hardly any connections in the graph. The figure below shows that for the whole graph, and subgraph taken from the entire network.

**Graphical user interface, text, application, email

Description automatically generated**

Figure 8.7: Average clustering

**Visualization and Exporting data for the app:**

At last, we visualize a sub graph from our large network, we tried to visualize the whole graph, but it was just a huge black block. Our graph is mostly with isolated nodes, in the subgraph we tried to take the nodes that have links, also we filtered them that the node with higher degree to be displayed as a larger node the figure below illustrate that:

**A picture containing text

Description automatically generated**

Figure 8.8: visualizing the subgraph.

As a conclusion: we wanted to relate this with our app, so we filtered the most cited books, and exported them with their attributes such, title and author as csv file, and they will be displayed in the app in the home page**.**

**Our data:** [Citation-network V1](https://lfs.aminer.cn/lab-datasets/citation/citation-network1.zip):  629,814 papers and >632,752 citation relationships (2010-05-15).

Our code: attached with the zip folder in the project submission in folder named: ‘Grad\_project\_python’