Clustering of Scientific Articles

Assignment for the course of Network Theory, ECE AUTh
Odysseas Sofikitis, 10130, sodyssea@ece.auth.gr, GitHub Repo

I. Introduction

A collection of 20K scientific articles was given, each categorized into one or more of the following topics: *Computer Science*, *Physics*, *Mathematics*, *Statistics*, *Quantitative Biology* and *Quantitative Finance*. An *NLP* model [1] was used to construct a graph, where each article is represented by a node, and edges were added based on the similarity scores of the NLP model. The goal of the assignment is to approximate the actual clusterings derived from the topic of each article with the clusterings of the NLP graph.

II. Creation Of The Graph

A) Data Sampling

The amount of articles in each topic is not uniformly distributed. Articles of *Quantitative Finance* or *Biology* are rare (~200-500), while topics like *Computer Science* more prevalent. If a proportional sampling was implemented, the clusters would be too small for those topics, while all other ones would be relatively big. Due to the use of the modularity algorithm, connections inside small clusters would have to be especially dense, while connections with other clusterings would have to be sparse, if any.

For the reasons above, an equal number of articles from each topic is selected. This is easily implemented by using a sampling size divisible by 6 (the number of topics) and keeping the articles in a new dictionary. It should be noted that this uniform sampling can be achieved only for samples up to 1.2K size. For samples of larger size, rare topics are sampled until exhausted and all other topics are sampled normally.

e. g. *For a sample size of 1.2K, 200 articles from each topic are chosen. On the contrary, for a sample of 6K, 1K articles of Computer Science, Physics, Mathematics and Statistics are chosen, while only 447 of Quantitative Biology and 209 of Quantitative Finance.*

B) Edges Generation

The main design parameter was to determine the conditions for an edge to be added. The *NLP* model does not provide immediate comparison of two texts, but their *vector embeddings* [2]. These embeddings are then used to calculate the similarity between the two texts. Initially, cosine distance and similarity was used for comparison. However, this method proved to result in too high values between all kinds of articles, making it more

difficult to decide on a cut-off value for the addition of an edge. After trial and error, common *Euclidean Distance* [3] was used instead, and an edge is added when similarity is greater than 0.46. It was also found that adding a weight proportional to the inverse of similarity improved the clustering results.

C) Clustering Distance

Following the creation of the graph, clustering is done using *best_partition* of *python-louvain* [4], which implements modularity detection. True clusterings of the data set are made for comparison, solely based on the data set, without any use of graphs. The two clusterings are then compared using distance of *Basic Clustering Matrices* [5].

III. Results

The results are poorer than expected. On average, clustering distance is equal to ⅓ of the total articles used. That is, for a sample of 0.5K, distance is equal to 180. On the other hand, the method shows good expendability. The ratio of nodes to distance is kept constant for larger samples, such as 1.2K, and performs almost the same even when the sample gets unbalanced due to exhaustion of some topics of the articles. It is worth mentioning that larger samples sometimes perform better with slightly different cut-off values and the results are non-deterministic due to the clustering algorithm used. *Figures 1* to *3* depict the graph for 0.5K nodes and different parameters.

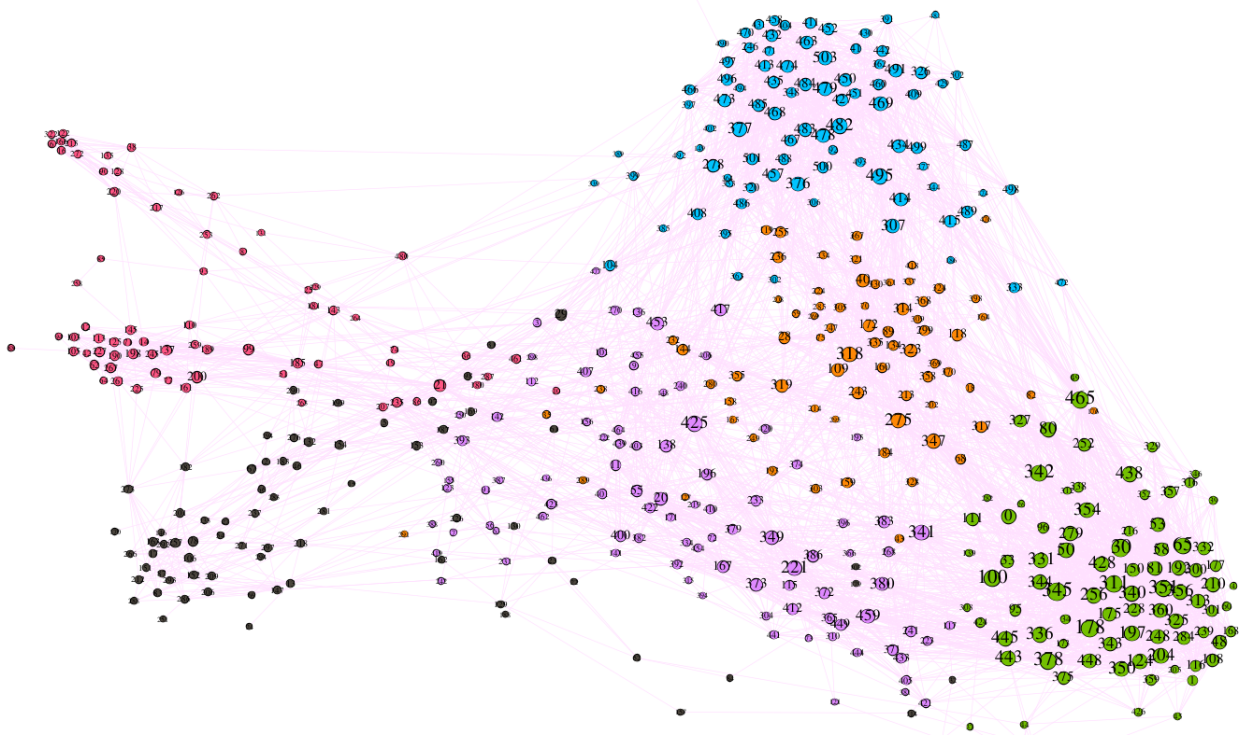

Figure 1. Graph of 504 nodes and ~5K edges, cut-off value of 0.46

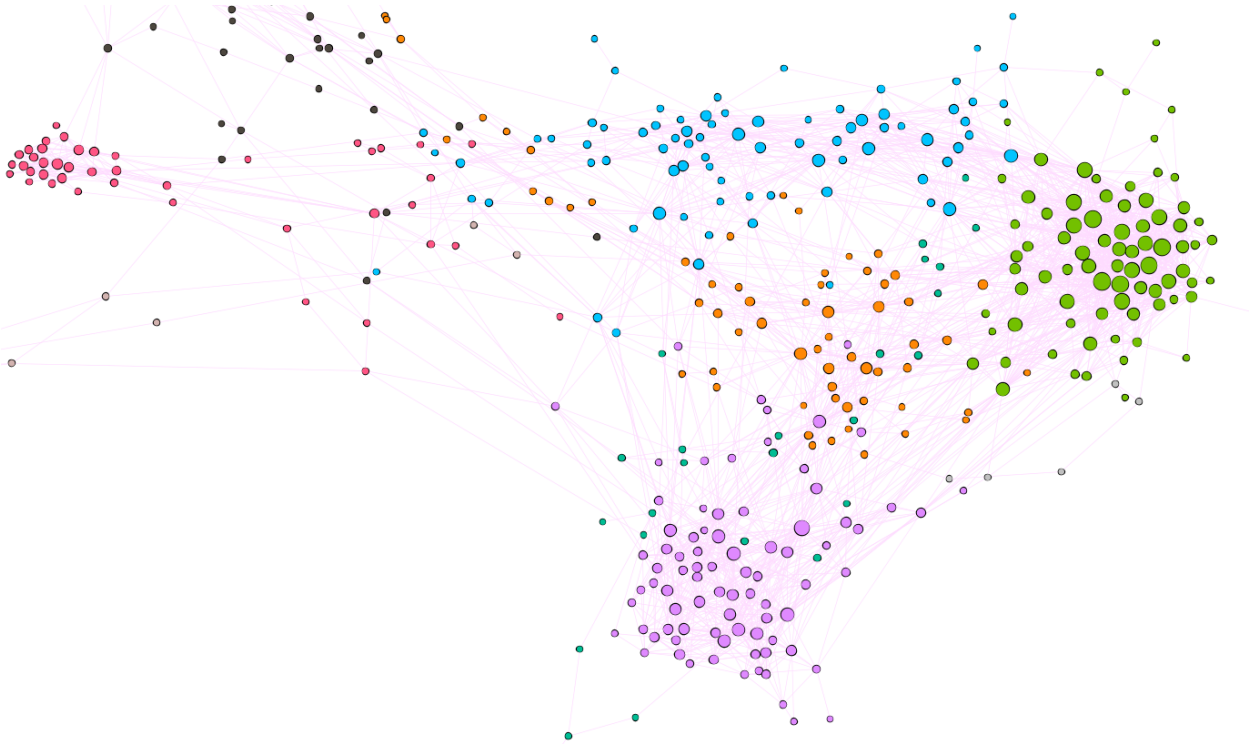

Figure 2. Graph of 504 nodes and ~2.5K edges, cut-off value of 0.47

Comparing *Figures 1* and *2* we can see that a 0.01 change in the cut-off value for adding edges can reduce the number of edges by up to half. Clusters in *Figure 2* have twice the distance from the optimal subgraph (*Fig 3)* than the clusters of *Figure 1*.

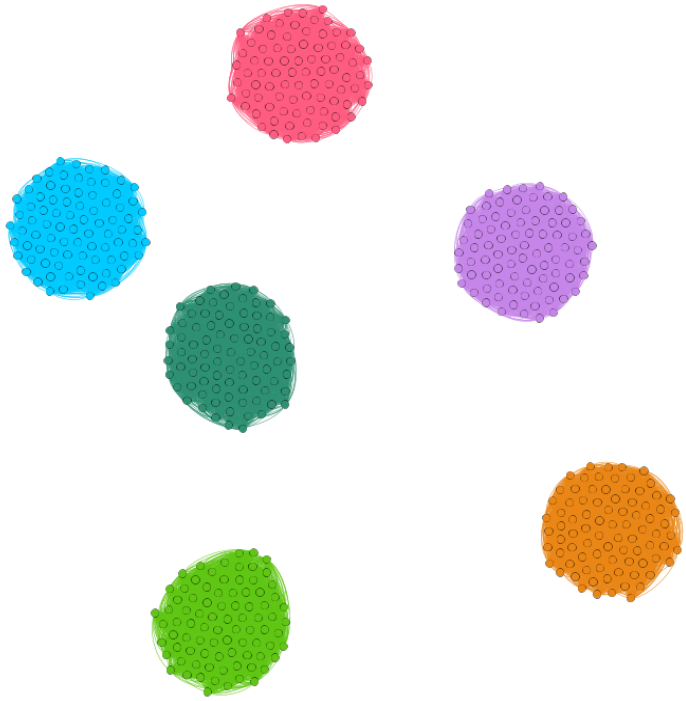

Figure 3. True depiction of the graph for 0.5K nodes

On *Figure 3* the best case scenario is shown, where only nodes that represent the same topic are connected with one another and the graph is separated into 6 fully connected subgraphs. These figures make clear the difficulty of approaching both the number of edges and the right connections of these edges through a simple cut-off value based on the similarity of the two texts.

IV. References

[1] Huggingface sentence transformers

[2] Vector Embeddings

[3] Euclidean Distance vs Cosine Similarity

[4] python-louvain

[5] Basic Clustering Matrices